

O3 – Obrazovni sadržaji za nastavnike

ZBIRKA SCENARIJA UČENJA ZA UČENJE PROGRAMIRANJA POMOĆU IZRADE IGARA

Podaci o dokumentu

Rezultat: O3/A1 - Zbirka scenarija učenja za učenje programiranja pomoću izrade igara

Broj i naslov intelektualnog rezultata: O3 – Obrazovni sadržaji za nastavnike

Odgovorni partner: Jugozapadno sveučilište “Neofit Rilski” (Bulgarska)

Partneri koji su sudjelovali u izradi: Sveučilište u Ljubljani (Slovenija), Sveučilište u Rijeci (Hrvatska)

Izjava o odricanju od odgovornosti

Projekt CODING4GIRLS je financiran iz programa Erasmus + Europske unije.

Ova publikacija sadrži podatke i izražava stajalište isključivo njenih autora i ne odražava nužno službeno mišljenje Europske unije. Institucije i tijela Europske unije niti bilo koja osoba koja djeluje u njihovo ime ne mogu biti odgovorni za uporabu informacija koje se nalaze u ovom dokumentu.

Coding4Girls, 2018-2020



Creative Commons Attribution-ShareAlike 4.0

International Public License ([CC BY-SA 4.0](#))

SADRŽAJ

1. UVOD	4
2. SCENARIJI UČENJA	5
OSNOVNI SCENARIJI UČENJA	7
<i>Scenarij učenja 1 – Uvod u sučelje alata Snap!</i>	<i>7</i>
<i>Scenarij učenja 2 – Vrijeme je za oživljavanje vašeg objekta</i>	<i>11</i>
<i>Scenarij učenja 3 – Kretanje po pozornici.....</i>	<i>14</i>
<i>Scenarij učenja 4 – Mijenjanje kostima i okretanje.....</i>	<i>20</i>
<i>Scenarij učenja 5 – Zvukovi s farme</i>	<i>25</i>
<i>Scenarij učenja 6 – Kameleonov ljetni odmor</i>	<i>33</i>
<i>Scenarij učenja 7 – Pomaganje princu i princezi u pronađasku njihovih životinja.....</i>	<i>39</i>
<i>Scenarij učenja 8 – Crtanje s kredom</i>	<i>44</i>
<i>Scenarij učenja 9 – Skupljanje otpadaka i čišćenje parka</i>	<i>53</i>
<i>Scenarij učenja 10 – Hranjenje mačaka</i>	<i>60</i>
<i>Scenarij učenja 11 – Pogađanje broja mačaka u skloništu</i>	<i>66</i>
NAPREDNI SCENARIJI UČENJA.....	71
<i>Scenarij učenja 12 – Hvatanje zdrave hrane</i>	<i>71</i>
<i>Scenarij učenja 13 – Pričam ti priču</i>	<i>78</i>
<i>Scenarij učenja 14 – Crtanje</i>	<i>85</i>
<i>Scenarij učenja 15 – Uhvati miša</i>	<i>85</i>
<i>Scenarij učenja 16 – Kupnja hrane za piknik</i>	<i>98</i>
<i>Scenarij učenja 17 – Operacije</i>	<i>105</i>
<i>Scenarij učenja 18 – Recikliranje</i>	<i>111</i>
<i>Scenarij učenja 19.1 – Sviranje klavira</i>	<i>116</i>
<i>Scenarij učenja 19.2 – Sviranje klavira</i>	<i>119</i>
<i>Scenarij učenja 20 – Test</i>	<i>128</i>
<i>Scenarij učenja 21 – Pojednostavljena igra PACMAN</i>	<i>134</i>
REFERENCE	139



1. UVOD

Vodeći psiholog prošlog stoljeća identificirao je igru kao jednu od najvažnijih aktivnosti za razvoj važnih životnih vještina, bez obzira na dob ili stadij razvoja. Dijete se kroz igru brzo prilagođava i snalazi novim okolnostima. Kad se igra, dijete otkriva osnovne pojmove iz stvarnog života i stvara odnose među njima.

Danas se igre sve češće koriste u najranijim fazama djetetova razvoja, i kod kuće i u vrtiću. Učenje u školi se još uvijek prečesto temelji na tradicionalnom prenošenju znanja modelom usmjerjenog na učitelja s pasivnim učenicima. S druge strane, teorije učenja razvijene u prošlom stoljeću, promoviraju nove pristupe učenju i poučavanju koji su usmjereni na učenike, temeljeni na problemima, usmjereni prema obrazovnim ishodima viših razina, motivirajući i često podržani informacijsko-komunikacijskom tehnologijom (IKT).

Pristupom CODING4GIRLS potiče se sudjelovanje u aktivnostima učenja programiranja stvaranjem vlastitih igara koje se odnose na probleme iz stvarnog svijeta. Pritom se učenicima predstavlja djelomično rješenje, a onda ih potiče da ga dovrše izradom manjih dijelova. Započinje se s rješavanjem lakših problema, a zatim se postepeno uvodi teže kako bi se potaknulo kontinuirano sudjelovanje i zadržavanja interesa kod učenika.

Kako bi se učiteljima pomoglo da integriraju CODING4GIRLS pristup u svoju nastavnu praksu, osmišljene su aktivnosti učenja: od osnovnih koje uvode samo jedan programski koncept do naprednijih s više programske koncepcije. Za aktivnosti učenja koje trebaju izgraditi vještine programiranja kod djevojčica i dječaka, pripremljeni su scenariji u skladu s CODING4GIRLS pristupom temeljenim na igrami. Pritom je fokus na identificiranim karakteristikama igara koje preferiraju djevojčice te onima koje uključuju probleme iz stvarnog života.

Svaki scenarij učenja uključuje sljedeće informacije:

- opći obrazovni cilj za aktivnost učenja
- pojmove obuhvaćene aktivnošću učenja
- specifične odgojno-obrazovne ciljeve
- očekivane ishode učenja
- opis koraka potrebnih za realizaciju CODING4GIRLS pristupa učenju temeljenog na izradi igara
- metode ocjenjivanja za vrednovanje usvojenog znanja
- pitanja za poticanje diskusije među učenicima prilikom suradnje u razredu.

Pripremljen je 21 scenarij s aktivnostima učenja. Učitelji mogu koristiti scenarije i igre u predloženom slijedu ili ih slobodno odabrati prema svojim željama i potrebama. Scenariji pokrivaju opću funkcionalnost predložene obrazovne igre, uključujući procese interakcije s korisnicima i davanje povratnih informacija, kao i opise svih aktivnosti učenja koje će se implementirati u predloženoj igri.

Scenariji učenja su dostupni na engleskom kao i na nacionalnim jezicima projektnih partnera - bugarski, hrvatski, grčki, talijanski, portugalski, slovenski i turski.



2. SCENARIJI UČENJA

Pripremljeni scenariji učenja predviđaju učenje jednog ili više koncepata i grupirani su u osnovne i napredne scenarije. Sljedećom tablicom je prikazan predloženi redoslijed izvođenja aktivnosti.

OSNOVNI SCENARIJI UČENJA		
1.	Uvod u sučelje alata Snap! Upoznavanje sa sučeljem alata Snap! Za vizualno programiranje	UL
2.	Vrijeme je za oživljavanje vašeg objekta Pronalaženje blokova koda i njihovo spajanje, pomicanje objekta, omogućavanje da objekt nešto kaže	UL
3.	Kretanje po pozornici Izrada smislenog slijeda blokova	UL
4.	Mijenjanje kostima i okretanje	UL
5.	Zvukovi s farme Dodavanje, uvoženje, snimanje i reproduciranje zvuka	UL
6.	Kameleonov ljetni odmor Upoznavanje s događajima, očitavanje boje, logičke vrijednosti, provjeravanje i reagiranje na dva različita stanja igre	UL
7.	Pomaganje princu i princezi u pronalasku njihovih životinja Korištenje uvjeta, crtanje	UL
8.	Crtanje s kredom Korištenje petlji, okretanje, promjena pozadine	UL
9.	Skupljanje otpadaka i čišćenje parka Uvod u varijable, dupliciranje objekata, blokovi koda	UL
10.	Hranjenje mačaka Korištenje varijabli (unutar i izvan petlji), petlje, spajanje stringova, slučajni brojevi, operatori, ulazne vrijednosti	UL
11.	Pogađanje broja mačaka u skloništu Korištenje slučajnih vrijednosti, unosa varijabli, uvjeta, operatora usporedbe, brojača	UL
NAPREDNI SCENARIJI UČENJA		
12.	Hvatanje zdrave hrane Korištenje varijabli, uvjeta, petlji, kretanje u smjeru, slučajne vrijednosti	UL
13.	Crtanje	UNIRI
14.	Pričanje priča	SWU



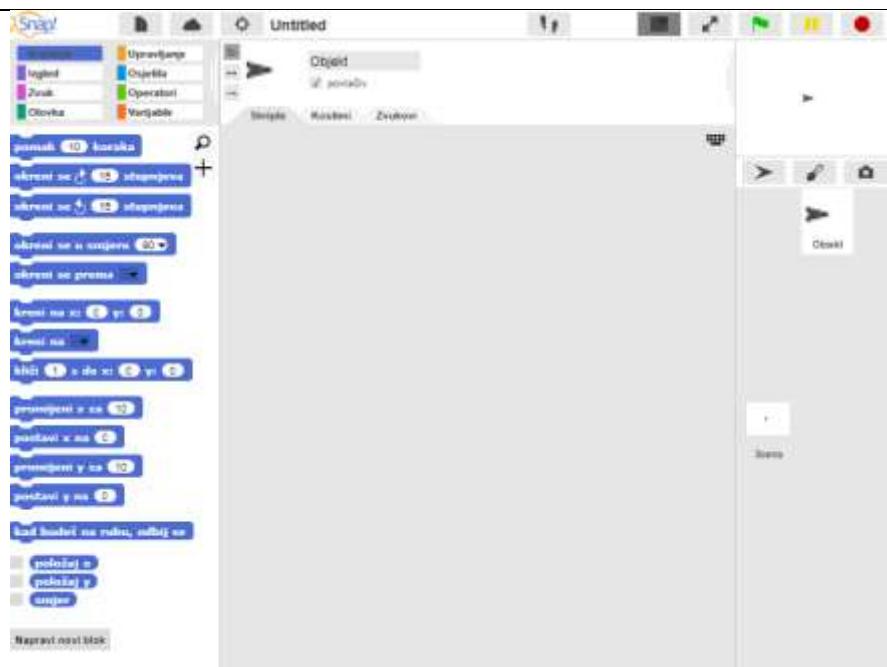
15.	Uhvati miša Petlje, uvjeti, varijable	UL
16.	Kupnja hrane za piknik Varijable, uvjeti, operatori	UL
17.	Operacije	SWU
18.	Recikliranje	SWU
19.1	Sviranje klavira 1	SWU
19.2	Sviranje klavira 2	UNIRI
20.	Test	SWU
21.	Pojednostavljena igra PACMAN Korištenje kretanja objekta temeljeno na događajima, očitavanje boje, logičke vrijednosti, provjeravanje i reagiranje na dva različita stanja igre	UL



OSNOVNI SCENARIJI UČENJA

Scenarij učenja 1 – Uvod u sučelje alata Snap!

Naziv scenarija	Uvod u sučelje alata Snap!
Potrebno predznanje iz programiranja	/
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">• objekti i pozadine u vizualnom programskom okruženju Snap! Ishodi učenja: <ul style="list-style-type: none">• Učenik će moći dodati novi objekt• Učenik će moći dodati kostime i urediti ih• Učenik će moći centrirati objekt, kako bi rotacija radila na ispravan način• Učenik će moći dodati različite pozadine na pozornicu i uređivati ih
Cilj, zadaci i kratki opis aktivnosti	Učenik dodaje nove objekte, dodaje kostime objektima, uređuje kostime te ih briše. Učenik stvara nove pozadine na pozornici, uređuje ju, te neželjene briše. Cilj: Do kraja sata učenici će nacrtati svoj omiljeni objekt i okruženje u kojem živi, stvarno ili imaginarno, kako bi ga mogli koristiti u igrici. Kako bi aktivnost bila motivirajuća za sve učenike, crtež objekta mora biti prikidan za ciljanu skupinu.
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Demonstracija Samostalni rad
Oblici poučavanja	Frontalni rad Individualni rad
Razrada aktivnosti	Do kraja sata učenici će nacrtati svoj omiljeni objekt i okruženje u kojem živi, stvarno ili imaginarno, kako bi ga mogli koristiti u igrici. [Korak 1] Učenicima je potrebno navesti web stranicu na kojoj mogu pronaći program Snap! (https://snap.berkeley.edu/). Pokazuju se različiti dijelovi sučelja: odjeljak s blokovima, odjeljak u kojem mogu sastavljati skripte, mijenjati kostime, dodavati zvukove, pozornicu sa objektima, listu objekata.



[KORAK 2]

Možete kreirati novi objekt klikom na jedan od sljedeća tri gumba:



Pokušat ćete nacrtati novi objekt, stoga ako kliknete na kist, otvorit će vam se prozor u kojem možete nacrtati svoj objekt na sličan način kao i u Paint-u.

Zadatak za učenike: Nacrtajte svoj prvi objekt. Imate 10 minuta. Nakon što nacrtate navedeni objekt, trebali biste se uvjeriti da je centar rotacije objekta onakav kakav želite da bude. Kako biste to učinili koristite



Zadatak za učenike: Centrirajte svoj objekt.

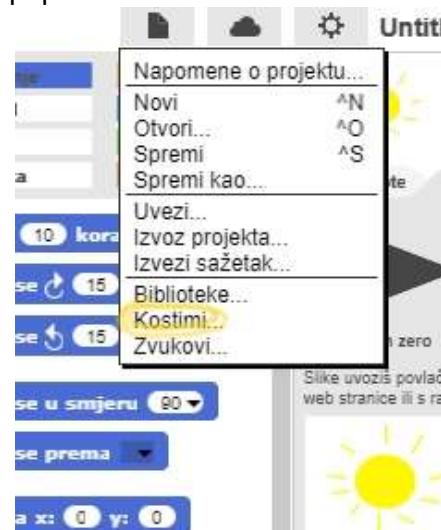
[KORAK 3]

Kako biste uredili svoj objekt, odaberite karticu Kostimi koja je vidljiva samo kada je objekt označen. Desnom tipkom miša kliknite na željeni kostim i odaberite uredi. Također, u istom izborniku možete duplicirati kostim ili ga izbrisati.



[KORAK 4]

Da biste umetnuli već postojeći kostim, kliknite na ikonu na kojoj je nacrtan komadić papira i odaberite Kostimi...



Ova opcija će se prikazati tek kada je vaš objekt označen na pozornici (sceni).

Zadaci za učenike: odaberite kostim i dodajte ga objektu

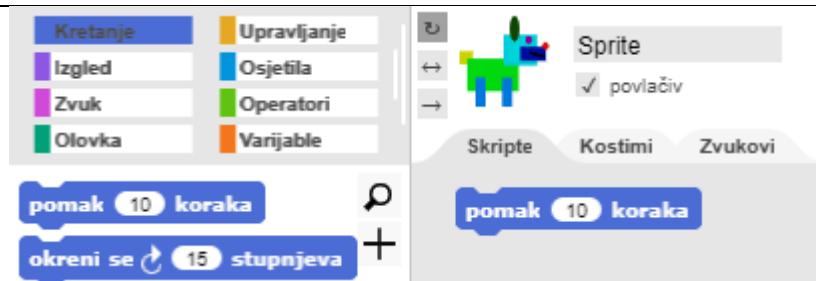


	<p>[KORAK 5]</p> <p>Sada kad imate svoj lik morate dodati neku pozadinu na scenu. Kako bi to napravili najprije trebamo kliknuti na Scena, a ne na objekt. Za dodavanje nove pozadine odabiremo karticu Kostimi:</p>  <p>Zadatak za učenike: Nacrtajte željenu pozadinu. Zadatak za učenike: Pretražite postojeće pozadine i dodajte jednu tako da ju uvezete, tako da imate dvije. Zadatak za učenike: Pronađite način na koji ćete urediti svoju pozadinu. Pronađite način brisanja pozadine tako da ostane samo jedna.</p> <p>Refleksija i evaluacija: Jesu li studenti uspijeli nacrtati svoj objekt i okruženje u kojem živi? Jesu li imali problema? Kako su ih riješili?</p>
Alati i materijali za nastavnike	Alat Snap!: https://snap.berkeley.edu/
Alati i materijali za učenike	/



Scenarij učenja 2 – Vrijeme je za oživljavanje vašeg objekta

Naziv scenarija	Vrijeme je za oživljavanje vašeg objekta
Potrebno predznanje iz programiranja	/
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">• slijed• objekt (sprite) Ishodi učenja: <ul style="list-style-type: none">• učenik će moći pronaći odgovarajuće programske blokove i povezati ih u niz• učenik će moći pomicati objekt• učenik će moći omogućiti da objekt nešto kaže
Cilj, zadaci i kratki opis aktivnosti	Učenik otkriva gdje su pohranjeni programski blokovi i kako pronaći odgovarajuće programske blokove, koje su kategorije blokova i kako se blokovi povezuju u niz.
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Demonstracija Samostalni rad
Oblici poučavanja	Frontalni rad Individualni rad
Razrada aktivnosti	<p>Učenicima se prezentira problem: Omogućit ćete da se vaš lik pomakne i nešto kaže tijekom ovog sata. Učenicima se može pokazati primjer programa koji će programirati na ovom satu.</p> <p>[Korak 1]</p> <p>Najprije pogledajmo gdje se nalaze programski blokovi koji su Vam dostupni za korištenje. Gdje se oni nalaze?</p> <p>Na lijevoj strani možete pronaći različite kategorije blokova: Kretanje, Izgled, Zvuk, Olovka, Upravljanje, Osjetila, Operatori i Varijable. Najprije ćemo koristiti pomak koraka blokove.</p> <p>Zadatak za učenike: Najprije pronađite blok, a zatim dvaput kliknite na njega. Što se dogodilo?</p> <p>[Korak 2]</p> <p>Da biste započeli programirati, trebate povući i ispustiti pomak koraka blokove u kartici Skripte.</p>



Dvostrukim klikom na blok u kartici Skripte omogućujete izvršavanje koda.

[Korak 3]

Programi u Snap!-u uobičajeno se pokreću klikom na zelenu zastavicu.

Zadatak za učenike: Klikom na različite kategorije blokova pokušajte pronaći blok koji omogućuje pokretanje programa ako se klikne zelena zastavica.

Rješenje:



Ako želite da program radi u ispravnom slijedu koraka, blokovi moraju biti povezani kao kod slagalica (*puzzle*).

Na primjer ovako:



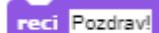
Sada će se, svaki put kada kliknete na zelenu zastavicu, objekt pomaknuti za 10 koraka, samo s drugog položaja na slici.

[Korak 4]

Ako se u bloku nalazi bijeli prostor, to znači da možete promijeniti brojeve ili slova koji su tamo upisani.

Zadatak za učenike: Omogućite da se vaš lik pomiče za 30 koraka odjednom, a umjesto samo 10.



	<p>[Korak 5]</p> <p>Omogućite da Vaš lik nešto kaže. Gdje ćete potražiti blok s naredbom „reci“? Isprobajte koja je razlika između  i  te objasnite razliku susjedu u klipi.</p> <p>[Korak 6]</p> <p>Pronašli ste obje „reci“ naredbe u kategoriji Izgled. Glavna razlika je u tome da s  Vi ne kažete programu da pričeka sekundi prije nastavka provođenja koda ili da bi ga trebao prestati izgovarati u bilo kojem trenutku.</p> <p>[Korak 7]</p> <p>Uzmite svoj lik s prethodnog sata. Povlačenjem na pozornicu, pomaknите ga na lijevu stranu pozornice i napišite program koji omogućuje liku  s njegove na lijevoj strani na desnu stranu pozornice. Nakon svakog pomaka lik bi trebao nešto reći. Napravite više od samo jednog pomaka.</p> <p>Isprobajte. Završi li lik na točno istom mjestu svaki put kada se program pokrene? Možete li pronaći blok koji bi osigurao da Vaš lik uvijek kreće s iste pozicije i da ne pobegne s pozornice?</p> <p>Savjet za nastavnika: Ako lik pobegne s pozornice, možete ga vratiti na pozornicu klikom na njega desnom tipkom miša i odabirom opcije „Pokaži“.</p> <p> Blok koji tražite je . Da biste odredili koje vrijednosti x i y su poželjne, možete pomaknuti svoj lik na mjesto na kojem želite da bude i klikom na položaj x i položaj y (na dnu kategorije blokova Kretanje) će se pokazati trenutne vrijednosti x i y. Vi ih samo morate upisati u bijela polja u bloku.</p> <p>Razmišljanje i evaluacija:</p> <p>Koliko puta Vaš lik mora ponoviti pokrete i izgovoriti nešto kako bi izvršio zadatak? Je li taj broj isti za sve u razredu? Zašto je to tako?</p>
Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_dog_goes_home
Alati i materijali za učenike	Ako učenik nije nacrtao svoj vlastiti objekt i pozadinu, može upotrijebiti: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_dog_goes_home_tmp

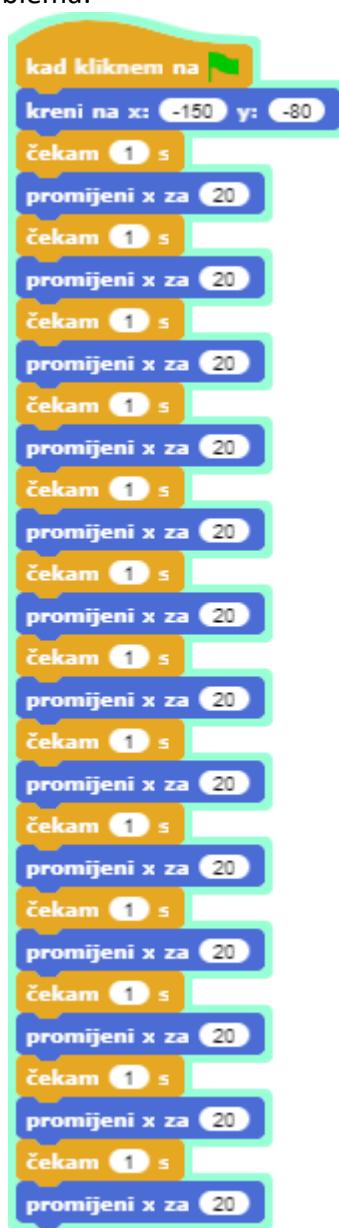


Scenarij učenja 3 – Kretanje po pozornici

Naziv scenarija	Kretanje po pozornici
Potrebno predznanje iz programiranja	Dodavanje novih objekata i pozadine Mijenjanje kostima Povezivanje blokova
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">Smisleno povezivati blokove Ishodi učenja: <ul style="list-style-type: none">Učenik će moći postavljati objekt na pozorniciUčenik će moći mijenjati x i y pozicije objektaUčenik će moći koristiti blok <i>ponavljam</i>Učenik će koristiti naredbu <i>pomak_koraka</i> za definiranje kretanja objekta u smjeru u kojem je objekt okrenut
Cilj, zadaci i kratki opis aktivnosti	Opis aktivnosti: Učenik će izvoditi kretanje objekta po pozornici u smjeru x i y, napraviti jednostavan program za izvršenje dobivenog zadatka, te mijenjati smjer objekta i uočiti utjecaj istog na blok <i>pomak_koraka</i> . Zadaci: Izraditi program u kojem će se objekt kretati u smjeru x, te izraditi program u kojem će se objekt kretati u smjeru y. Cilj: Učenik će razlikovati kretanja u smjeru x od kretanja u smjeru y te će koristiti petlju <i>ponavljam</i>
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Demonstracija Individualni rad
Oblici poučavanja	Frontalni rad Individualni rad
Razrada aktivnosti	Učenicima se prezentira problem: Različitim životinjama je potrebno pomoći u ostvarenju njihovih ciljeva. Kako bi se to ostvarilo, treba ih uputiti kako se gibati po pozornici. [Korak 1] Otvori program <i>Uhvati loptu</i> i izmjeni kod kako bi pas uhvatio loptu. Koristi se naredbama promijeni x za i čekam kako bi se pas pomicao prema lopti.



Moguće rješenje problema:



Kako bismo pokrenuli ovaj blok naredbi, potrebno je kliknuti na zastavicu označenu zelenom bojom . Kao što se može primijetiti, vrijednost *x* se mijenja, što daje za rezultat da se objekt pas kreće u smjeru *x*.

Kada bismo postavili početnu vrijednost *x* na 0, tada bi se objekt pas nalazio na sredini pozornice. Ukoliko je vrijednost *x* postavljena na vrijednost koja je manja od 0 (kao u ovom primjeru), onda se objekt pas nalazi u lijevoj polovici pozornice. Ukoliko je vrijednost *x* postavljena na vrijednost koja je veća od 0, onda se objekt pas nalazi u desnoj polovici pozornice. Uočite da je u tom slučaju pas bliži lopti.



Savjet:

Ukoliko je se radi o radu sa starijim učenicima kojima su unaprijed poznati brojevi u decimalnom zapisu, vrijeme čekanja se može postaviti na kraći period, npr. 0.1. Ukoliko su učenici upoznati s koordinatnim sustavom, objašnjenje istog se može izbjegći.

[Korak 2]

Otvori *Pomozi majmunu da se popne na drvo* i izmjeni kod kako bi majmun uhvatio banane. Koristi se blokovima **promijeni y za** i **čekam** kako bi se majmun popeo na palmu.

Moguće rješenje problema:



Kako bismo pokrenuli ovaj blok naredbi, potrebno je kliknuti na zastavicu označenu zelenom bojom . Kao što se može primijetiti, vrijednost y se mijenja, što daje za rezultat da se objekt pas kreće u smjeru y.

Kada bismo postavili početnu vrijednost y na 0, tada bi se objekt pas



nalazio na sredini pozornice. Ukoliko je vrijednost y postavljena na vrijednost koja je veća od 0, onda se objekt pas nalazi u gornjoj polovici pozornice. Kako bismo objekt smjestili ispod sredine pozornice, možemo zamisliti da se objekt nalazi na površini mora ($y=0$) , te da je potrebno zaroniti u more pri čemu se spuštamo u dubinu i time odmičemo od površine mora. Nakon spuštanja u dubinu mora, glavno pitanje je koliko metara smo ispod površine mora, drugim riječima: koliko koraka smo ispod sredine pozornice. Za spuštanje objekta majmuna sa palme potrebno je koristiti blok **promijeni y za -10**.

Savjet:

Ukoliko je rad sa starijim učenicima kojima su poznati brojevi u decimalnom zapisu, vrijeme čekanja se može postaviti na kraći period, npr. 0.1. Ukoliko su učenici upoznati s koordinatnim sustavom, objašnjenje istog se može izbjegići.

[Korak 3]

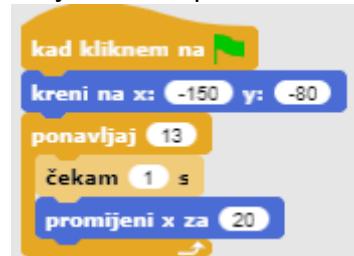
U oba zadatka koristila se kombinacija dvaju blokova. Koliko puta ste **ponavljali isti kod?**

Postoji kraći način zapisa istog koda tako da se kompjuteru kaže da isti kod ponovi određeni broj puta. Radi se o petlji **ponavljam__**. Petlja **ponavljam__** se može koristiti ako se jedna radnja ili niz radnji ponavlja više puta. U oba zadatka izmijeni kod koristeći se petljom



. Dio koda koji želiš ponavljati je potrebno smjestiti unutar bloka **ponavljam__**, i potrebno je zapisati broj ponavljanja u prazni dio bloka.

Linije koda za psa:

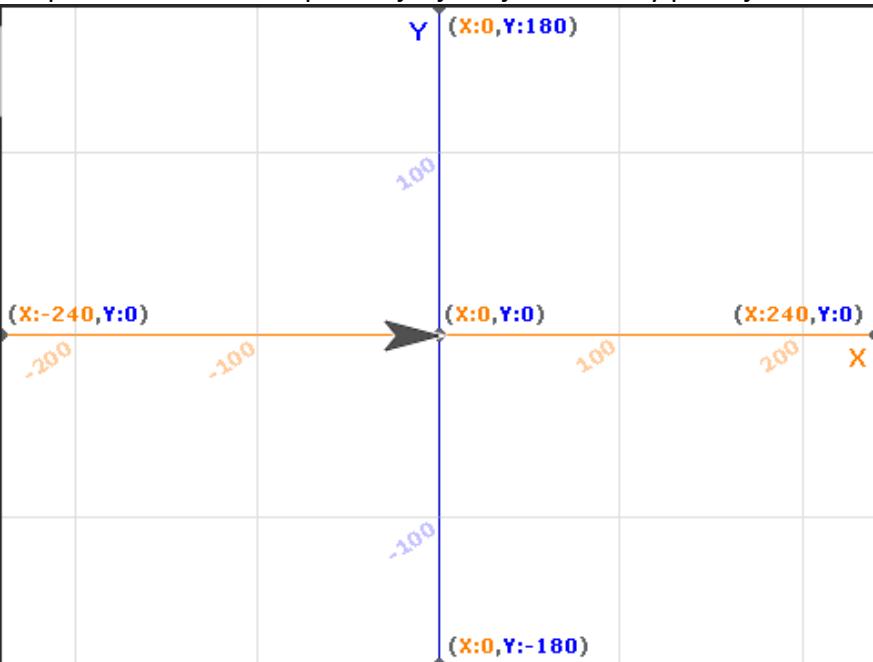
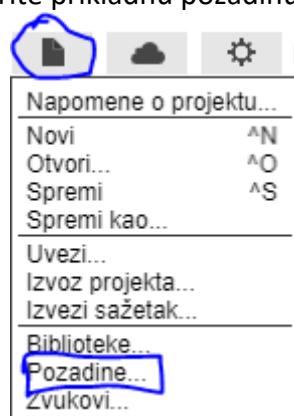


Linije koda za majmuna:



Zadatak: Pokušaj napraviti da pas trči k lopti i nazad.



	<p>Zadatak: Pokušaj napraviti da se majmun penje po stablu i zatim spušta.</p> <p>Što ti se najviše dopalo? Koristi XY pozadinsku mrežu u programu Snap kako bi si olakšao postavljanje objekta na x i y poziciju:</p>  <p>Za postavljanje XY pozadinske mreže, prvo se smjestite na pozornicu (ne na lika), kliknite na ikonu lista u lijevom gornjem kutu, odaberite <i>pozadine</i> i zatim izaberite prikladnu pozadinu pod nazivom XY grid.</p>  <table border="1"><tr><td>Alati i materijali za nastavnike</td><td>Moguće rješenje za <i>Uhvati loptu</i>: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_moving_x</td></tr><tr><td></td><td>Moguće rješenje za <i>Pomozi majmunu popeti se na drvo</i>: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_moving_y</td></tr></table>	Alati i materijali za nastavnike	Moguće rješenje za <i>Uhvati loptu</i> : https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_moving_x		Moguće rješenje za <i>Pomozi majmunu popeti se na drvo</i> : https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_moving_y
Alati i materijali za nastavnike	Moguće rješenje za <i>Uhvati loptu</i> : https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_moving_x				
	Moguće rješenje za <i>Pomozi majmunu popeti se na drvo</i> : https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_moving_y				



**Alati i materijali za
učenike**

Uhvati loptu:

https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_Catch_the_ball

Pomozi majmunu popeti se na drvo:

https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_Help_monkey_climb_the_tree

**Scenarij učenja 4 – Mijenjanje kostima i okretanje**

Naziv scenarija	Mijenjanje kostima i okretanje
Potrebno predznanje iz programiranja	Pomicanje
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Izrada smislenog slijeda blokova Ishodi učenja: <ul style="list-style-type: none">● Učenik će moći mijenjati kostim objekta kako bi napravio animaciju● Učenik će moći mijenjati rotaciju likova
Cilj, zadaci i kratki opis aktivnosti	OPIS AKTIVNOSTI: Učenik uči kako promijeniti kostur objekta te kako napraviti animaciju. Također uči kako se između njih može mijenjati različite vrste rotacije objekta. ZADACI: Stvoriti program koji mijenja kostim objekta te u svakom programu postaviti odgovarajuću vrstu rotacije za svaki objekt. CILJEVI: Moći promijeniti kostur objekta i postaviti prikladnu vrstu rotacije objekta.
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Game Based Learning – igra Razgovor Demonstracija Rješavanje problema Individualni zadatak
Oblici poučavanja	Frontalni rad Rad u paru Individualni rad Grupni rad (svi učenici)
Razrada aktivnosti	Učenici će naučiti kako napraviti animaciju objekta tako da izgleda kao hodanje, plesanje i slično. [KORAK 1] Otvorite novi prazan projekt, kliknite na ikonu koja izgleda kao bijeli komad papira i odaberite <i>Kostimi...</i> Kliknite na balerinu „A“ i zatim kliknite Uvoz (ili „Import“). Učinite isto s balerinom „B“, balerinom „C“ i balerinom „D“. Na kartici <i>Kostimi</i> vašeg objekta sada imate 4 kostima balerine. Možete preimenovati objekt u Balerinu tako da promijenite tekst iznad. Kartica kostima:

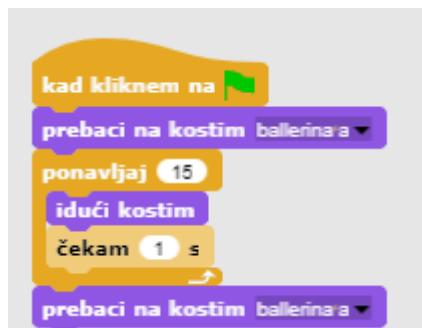


Vratite se na karticu Skripta i pokušajte stvoriti kod koji će početi kad se klikne na zelenu zastavu i 15 puta se mijenja pojava balerine

svake sekunde. Morat ćete koristiti blok **idući kostim**.

Pazite da balerina započne i završi ples s obje noge na podu. Početni i krajnji položaj nisu dio njenog plesa.

RJEŠENJE:



[KORAK 2]



Naša balerina ne želi cijelo vrijeme biti na istoj poziciji, pa radi male pokrete svaki put kad promijeni kostim. Dodajte ovaj pokret njenom plesu.

MOGUĆE RJEŠENJE:



[KORAK 3]

Otvorite novi prazan projekt i uvezite "Avery walking". Dodajte prigodnu pozadinu za Avery da krene dalje. Animirajte Avery tako da hoda s lijeve strane pozornice na desnu stranu pozornice. Pokušajte shvatiti kako animirati Avery tako da njezini koraci izgledaju povezani kao u stvarnom životu.

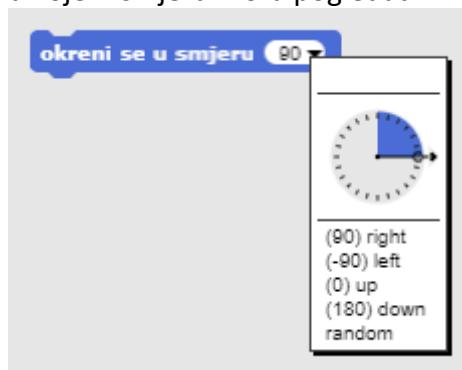
MOGUĆE RJEŠENJE:



[KORAK 4]

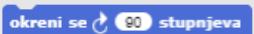
Do sada ste uvijek pisali program u kojem se objekt kretao samo u jednom smjeru. U ovom koraku morat ćete okrenuti miš kako bi došli do sira. Da bi ga okrenuli, možete odabrati sljedeće:

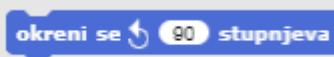
- a) naredbu u kojem smjeru mora pogledati



- b) ili možete definirati da se okreće za određeni kut u smjeru



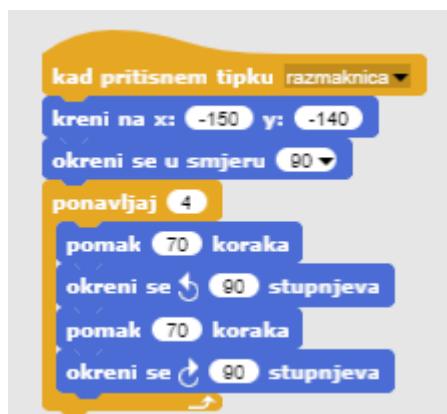
kazaljke na satu  ili u smjeru suprotnom

od kazaljke na satu 

Puni krug ima 360 stupnjeva, pa ako se želite okrenuti suprotno od mjesta gdje trenutno stojite, okrenite se za 180 stupnjeva. Ako želite skrenuti lijevo, skrenite za 90 stupnjeva kazaljke na satu. Ako želite skrenuti s desne strane, skrenite 90 stupnjeva u smjeru kazaljke na satu.

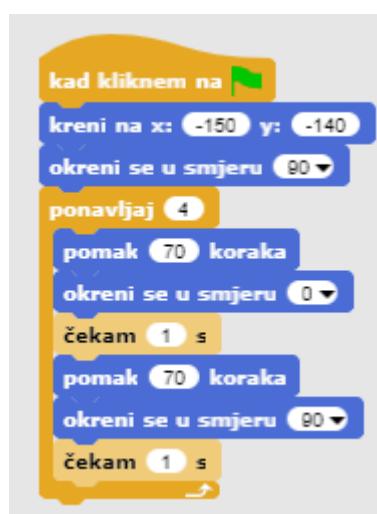
Otvorite <http://bit.do/fkEZc>. Napišite naredbe koje miš mora slijediti kako bi došao do sira ako mora hodati samo po zelenoj površini. Usmjerite miša u smjeru u kojem se kreće i pomaknite __ koraka. Da bi vidjeli kako se miš kreće, upotrijebite naredbu "pričekajte 1 sekundu između redova".

RJEŠENJE:

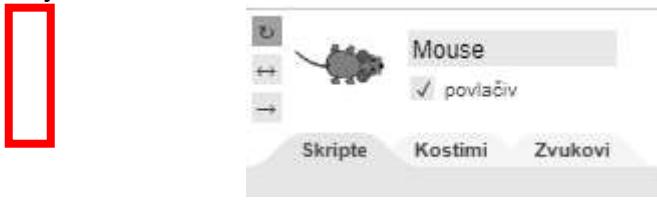


Sada pokušajte napisati program s okretanjem za 90 stupnjeva.

RJEŠENJE:





	<p>[KORAK 5]</p> <p>Kao što ste vidjeli, miš se okrenuo u različitim smjerovima kako bi došao do sira. Ponekad ne želite da se vaš objekt okreće naopako već želite da se samo okreće na lijevo ili desno tako da ne hoda po svojoj glavi. Da biste bili sigurni da se vaš objekt okreće onako kako želite, morate kliknuti na odgovarajuću ikonu lijevo od vašeg objekta:</p>  <p>Kružna strelica znači da se vaš objekt može okretati u bilo kojem smjeru (poput vašeg miša).</p> <p>Strelica <-> znači da će se vaš objekt okrenuti samo ulijevo ili udesno (ovo biste koristili za psa da ne hoda po glavi).</p> <p>Zadnja -> strelica znači da će objekt uvijek izgledati onako kako jest (ovo možete koristiti za majmune).</p> <p>Pokušajte napraviti svoje programe za psa i majmuna po uzoru na ovaj, tako da hodaju do predmeta i okrenu se natrag. Pripazite na primjenu pravilnog rotacijskog stila.</p>
Alati i materijali za nastavnike	<p>Igra izrađena alatom Snap!: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_dancing</p> <p>Avery hodanje: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_Avery_walking</p> <p>Pronađi sir - rješenje: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_Find_cheese_solution</p>
Alati i materijali za učenike	<p>Polugotova igra izrađena alatom Snap!: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_Find_cheese</p>



Scenarij učenja 5 – Zvukovi s farme

Naziv scenarija	Zvukovi s farme
Potrebno predznanje iz programiranja	Postavljanje pozadine Dodavanje novog objekta Omogućiti da objekt nešto kaže
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Dodavanje zvuka iz Snap!-ove medijske biblioteke,● Uključivanje zvuka iz drugih medija,● Snimanje novog zvuka,● Sviranje zvuka na pritisak tipke. Ishodi učenja: <ul style="list-style-type: none">● Učenik će moći dodati zvuk iz Snap!-ove medijske biblioteke i pustiti ga da svira pritiskom određene tipke na tipkovnici,● Učenik će moći dodati zvuk sa računala i pustiti ga da svira pritiskom određene tipke na tipkovnici,● Učenik će moći snimiti novi zvuk i pustiti ga da svira pritiskom određene tipke na tipkovnici.
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Programirati jednostavnu igru u kojoj igrač uči zvukove životinja pritiskanjem određenih tipki na tipkovnici. Zadatak: U prvom koraku učenik treba odabrati pozadinu scene. Zatim, učenik treba programirati farmericu koja govori upute: 1) Ako želiš čuti psa, pritisni tipku „D“!; 2) Ako želiš čuti svinju, pritisni tipku „P“!; 3) Ako želiš čuti kravu, pritisni tipku „C“!; 4) Ako želiš čuti ovcu, pritisni tipku „S“!; 5) Ako želiš čuti konja, pritisni tipku „H“!. Nakon toga, učenik treba programirati sviranje zvukova. Cilj: Učenici će se upoznati s načinom kako dodati novi zvuk i kako ga koristiti. Također će naučiti kako koristiti blok <i>Zvuk („odsviraj zvuk [naziv zvuka]“)</i> i blok <i>Upravljanje („kad pritisnem tipku [tipka]“)</i> .
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Na primjer: Game Based Learning – igra Aktivno učenje
Oblici poučavanja	Frontalni rad Individualni rad

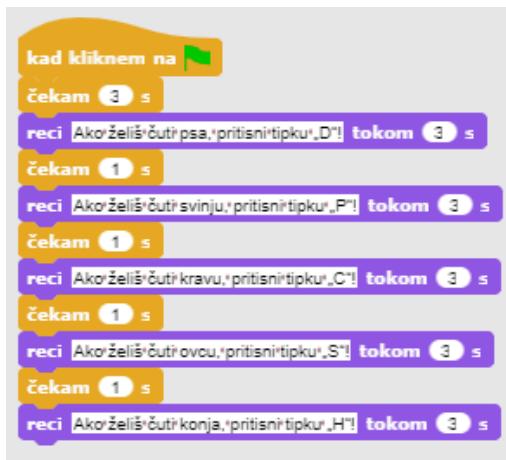


Razrada aktivnosti	<p>Motiviramo učenike igranjem igre (oni ne vide kod). Cilj je napraviti igru kao što je ova koju učenici igraju.</p>  <p>[Korak 1]</p> <p>Prvi korak je utvrđivanje pozadine igre. Pozadina treba sadržavati različite životinje. Postoje tri opcije:</p> <ol style="list-style-type: none">1. učenici sami crtaju pozadinu;2. učenici traže besplatnu sliku na mreži;3. učitelj pripremi pozadinu (zbog uštede vremena). <p>Učenici već znaju kako dodati pozadinu pa to rade samostalno.</p>  <p>[Korak 2]</p> <p>Drugi korak je dodati farmericu. Kao u prvom koraku, tri su opcije:</p> <ol style="list-style-type: none">1. učenici sami crtaju farmericu;2. učenici traže besplatnu sliku farmerice na mreži;3. učitelj pripremi sliku farmerice učenicima (ako želimo uštedjeti vrijeme). <p>Učenici već znaju kako dodati novi objekt pa to rade samostalno.</p>
--------------------	---



[Korak 3]

Nadalje, učenici trebaju programirati upute za igrača. Upute daje farmerica. Učenici to čine koristeći *Izgled/reci[znak]* i *čekam[n]* blok. Učenici već znaju kako to napraviti pa to izvršavaju samostalno.



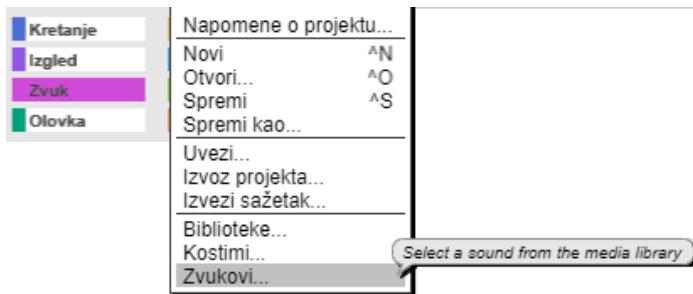
U nastavku pokažemo učenicima kako dodati zvuk u igru. Imamo tri opcije:

1. Dodavanje zvuka iz Snap!-ove medijske biblioteke;
2. Dodavanje zvuka s našeg računala povlačenjem u Snap!;
3. Snimanjem novog zvuka u Snap!-u.

Pokažemo učenicima sve tri opcije frontalnim načinom rada. Kada ih sve demonstriramo, sljedeće zadatke učenici počinju programirati samostalno (uz pomoć učitelja).

[Korak 4]

Učenici trebaju programirati zvuk psa. Kada igrač pritisne tipku "D", pas treba lajati. Prvo, učenici dodaju zvuk iz Snap! medijske biblioteke u pozadinsku karticu *Zvukovi*.

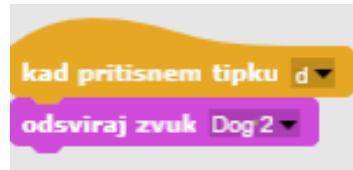




Zatim, biraju zvuk psa (Dog 1 ili Dog 2).



Učenici trebaju programirati zvuk psa koji će svirati kada je pritisnuta tipka "D". To rade koristeći *Upravljanje/kada pritisnem tipku [tipka]* blok i *Zvuk/odsviraj zvuk [ime_zvuka]* blok.



[Korak 5]

Učenici trebaju programirati zvuk životinja. Prvo, trebaju dodati zvukove sa svojih računala. To čine povlačenjem zvukova u pozadinsku karticu *Zvukovi*.

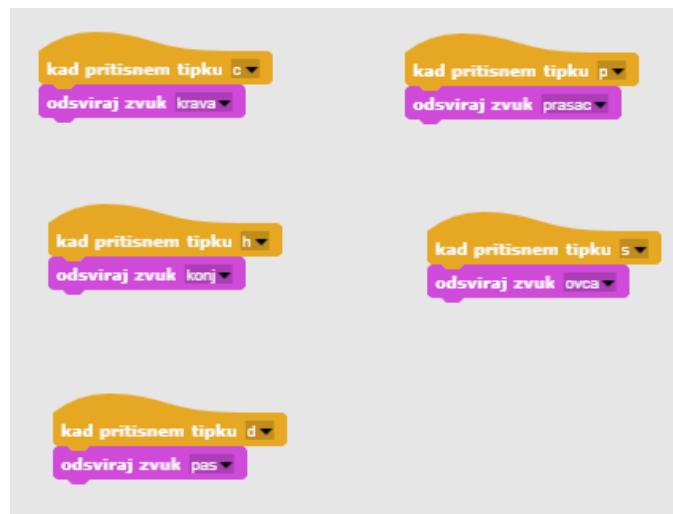


Jednom kad su dodani zvukovi, možemo ih preimenovati desnim



klikom na pojedini zvuk. U našem slučaju zvukovi se zovu krava, prasac, konj, ovca i pas.

Nadalje, učenici trebaju dodati zvukove u pozadinsku karticu *Skripte*. To čine koristeći *Upravljanje/kad pritisnem tipku [tipka]* i *Zvukovi/odsviraj zvuk [naziv_zvuka]* blok.



[Korak 6]

Idući korak je programirati farmeričin pozdrav dobrodošlice. Kada igrač započinje igru farmericu treba reći: "Dobro došli na moju formu". Prvo, učenici trebaju snimiti farmeričin pozdrav. To će učiniti pomoću snimača zvuka (crveni gumb) koji se nalazi u kartici *Zvukovi*. Kada snime zvuk, trebaju ga spremiti (gumb Spremi).

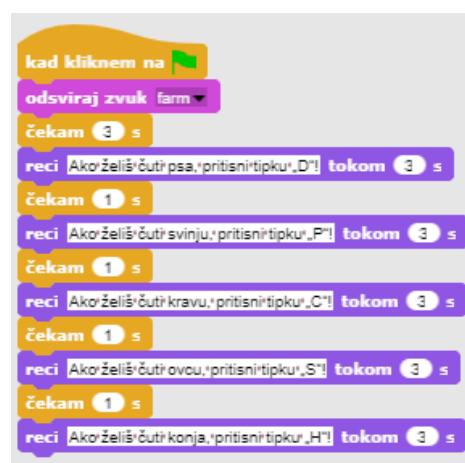


Jednom kad je zvuk spremljen, možemo ga preimenovati desnim klikom na njega.



Sada učenici trebaju dodati zvuk u skriptu farmerice. To čine koristeći *Zvuk/odsviraj zvuk [naziv_zvuka]* blok.

odsviraj zvuk [farm v1]



Refleksija i evaluacija

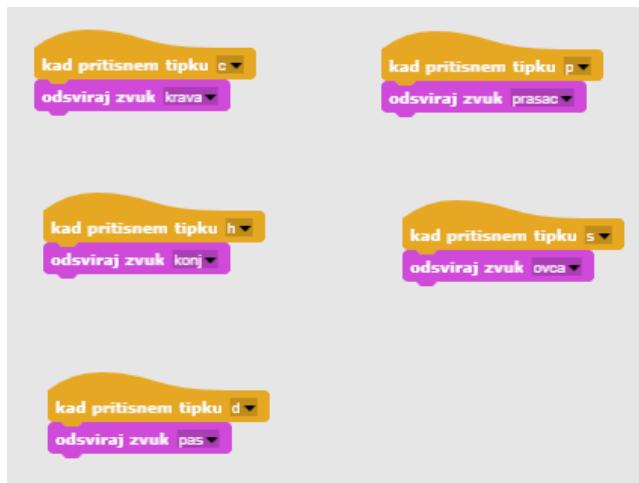
S učenicima se ponavlja:

- Kako su dodali zvukove u njihov kod.
- Koje su blokove koristili da bi dodali zvuk u kod.
- Koje upravljačke blokove su koristili u svom kodu.
- Zašto i kako su koristili blokove za zvuk i upravljačke blokove.

[Cijeli kod]
Farmerica



Pozadina



[Dodatni zadatak]

Učenici mogu doraditi igru tako što će dodati nove objekte (farmera, kokoš, traktor,...) i zvukove.

Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=tadeja&project=Farm Web-stranica za besplatno preuzimanje slika: https://pixabay.com/ Web-stranica za besplatno preuzimanje zvukova: https://www.zapsplat.com/ Lajovic, S. (2011). Scratch. <i>Nauči se programirati in postani računalniški maček</i> . Ljubljana: Pasadena. Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i> . Ljubljana: MK.
---	--



Alati i materijali za učenike	<p>Predložak izrađen u alatu Snap!:</p> <p>https://snap.berkeley.edu/project?user=tadeja&project=Sounds%20of%20the%20farm_0</p> <p>Web-stranica za besplatno preuzimanje slike: https://pixabay.com/</p> <p>Web-stranica za besplatno preuzimanje zvukove: https://www.zapsplat.com/</p>
--------------------------------------	--

**Scenarij učenja 6 – Kameleonov ljetni odmor**

Naziv scenarija	Kameleonov ljetni odmor
Potrebno predznanje iz programiranja	Dodavanje i uređivanje pozadine i lika Omogućiti da lik govori Korištenje naredbe <i>Ako ... onda ...</i> Kretanje objekta pomoću određenih tipki (npr. strelica)
Ishodi učenja	<p>Koncepti programiranja:</p> <ul style="list-style-type: none">• kretanje objekta na temelju događaja• očitavanje jedne ili više boja• čitanje logičke vrijednosti u logičkim izrazima• definiranje, razlikovanje, dinamičko provjeravanje i reagiranje na različita stanja igre <p>Ishodi učenja:</p> <ul style="list-style-type: none">• učenik će moći omogućiti kretanje objekta korištenjem tipki sa strelicama, koristeći događaje i uzima u obzir ograničenja,• učenik će moći koristiti blok osjetila u boji za dobivanje logičke vrijednosti pri očitavanju jedne ili više boja,• učenik će moći uočiti da se stanje objekta može mijenjati u ovisnosti boja koje dodiruje,• učenik će moći razlikovati pet različitih stanja i zna kako ih izraziti logičkim izrazima,• učenik će moći uočiti da se položaj objekta dinamički mijenja i koristi zauvijek petlju za ponovnu provjeru trenutnog stanja,• učenik će moći koristiti naredbe <i>Ako ... onda ...</i> za različite odgovore na temelju trenutnog položaja objekta.
Cilj, zadaci i kratki opis aktivnosti	<p>Kratki opis: Programirajte jednostavnu igru u kojoj će objekt promijeniti svoj kostim na temelju boje pozadine.</p> <p>Zadaci: Učenici moraju programirati kameleona da promjeni svoj izgled (kostim) i reći gdje se on nalazi u pet različitih situacija: 1) kad pliva u moru, mora promijeniti boju u plavu i reći „Plivam u moru“, 2) kad se nalazi između mora i plaže, koža mu postaje pola plava, pola pješčane boje i kaže: " Nalazim se između mora i plaže ", 3) na plaži poprima pješčanu boju i kaže "Opuštam se na plaži", 4) između plaže i šume, poprima pola zeleno, pola pješčanu boju i kaže "Nalazim se između plaže i šume", 5) u šumi, koža mu postaje zelena i on kaže: „Hladim se u hladu drveća“.</p> <p>Cilj:</p>



	Učenici će biti upoznati s blokom naredbi osjeta boja i kako ga koristiti u logičkim izrazima da bi razlikovali dinamično promjenjiva stanja igre i dali prave odgovore.
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje Suradničko učenje Rješavanje problema
Oblici poučavanja	Frontalni rad Rad u paru Individualni rad Grupni rad (svi učenici)
Razrada aktivnosti	<p>Učenicima se prezentira problem: Kameleon je otišao na ljetni odmor. Voli se kupati u moru, uživati u opuštanju na plaži, a kad je previše vruće, voli ići u sklonište obližnjih stabala kako bi se rashladio. Budući da se radi o kameleонu, on treba mijenja boju u skladu s trenutnom pozadinom.</p> <p>[Korak 1]</p> <p>Učenici bi trebali urediti pozadinu scene tako da je podijeljena na tri jednakih dijela, a svaki predstavlja drugo mjesto: plava boja je za more, pješčana boja za plažu i zelena za šumu. Učenici mogu dodati druge stavke kako bi pozadinu učinili što realnijom poput valova, školjki, dvoraca s pijeskom, suncobrana, drveća itd., ali moraju biti oprezni da dodani predmeti nisu veći od samog glavnog lika, jer u tom slučaju kameleon neće dodirnuti nijednu od tri boje, a Snap-ova senzorna značajka neće moći prepoznati na kojem dijelu scene je lik.</p>  <p>[Korak 2]</p> <p>Učenici trebaju nacrtati kameleona i obojiti mu kožu u pet različitih kombinacija koje predstavljaju njegov položaj na sceni:</p>  <p>[Korak 3]</p> <p>Prvo učenici trebaju napraviti da se njihov kameleon okreće u četiri</p>



smjera pomoću tipki. Mogu odabratи vlastitu kombinaciju tipki (npr. tipke sa strelicama ili tipke W, A, S i D). U ovom trenutku pretpostavljamo da znaju kako to učiniti. Potrebno je upozoriti učenike da ne zaborave da se taj lik može maknuti sa scene ako ne koristimo odgovarajući blok pri programiranju pokreta (kameleon se treba odbiti ako je na rubu bloka). Da bi kretanje kameleona bilo malo realnije, želimo da skrene ulijevo ili udesno u vodoravnom smjeru u kojem se nalazimo (koristeći naredbu u bloku smjera).



[Korak 4]

Upoznajemo studente s konceptom osjeta boje koju lik dodiruje. Pomoću naredbe "Dodiruje?" možemo dobiti informacije u obliku logičnih vrijednosti - istina ili laž ako dotakne jednu ili čak više boja u isto vrijeme. Budući da iz ovog bloka dobivamo logičku vrijednost, možemo je upotrijebiti u uvjetu *Ako ... onda ...*- naredbe, gdje se odlučuje hoće li se izvršavati naredbe navedene u njegovom tijelu ili ne.

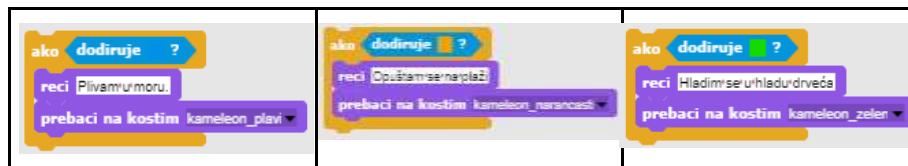
Zatim razgovaramo s učenicima koji su različiti položaji kameleona na sceni i kako ih možemo izraziti blokom naredbi „*Dodiruje?*“.

Brzo saznajemo da ih ima pet:

1. Kameleon je u potpunosti na plavom dijelu -> *Dodiruje [plava]?*
2. Nalazi se između plavog i pješčanog dijela -> *Dodiruje [plava]?* i *Dodiruje [pjesak]?*
3. On je u cijelosti na pješčanom dijelu -> *Dodiruje [pjesak]?*
4. Nalazi se između pješčanog i zelenog dijela -> *Dodiruje [pjesak]?* i *Dodiruje [zeleno]?*
5. On je u cijelosti na zelenom dijelu -> *Dodiruje [zeleno]?*

Kad kameleon dodirne određenu boju / boje, moramo promijeniti njegov izgled, te moramo napraviti da kaže gdje se nalazi. Izgled kameleona možemo promijeniti zamjenom njegovih kostima. To se postiže blokom *Izgled / prebaci na kostim [opcija]* gdje odabiremo koji od mogućih kostima želimo prikazati. Da bi kameleon govorio koristimo blok *Izgled / reci [tekst]*.

Prvo se pobrinemo za jednostavnije situacije kada je kameleon u potpunosti na istoj boji u sceni:





Dalje oblikujemo logički izraz upotrebom logičkog operatera *I*, jer želimo provjeriti dodiruje li kameleon dvije boje istovremeno:



Ako kombiniramo gore navedene uvjetne rečenice i stavimo ih pod „*Kad kliknem na Zelena zastava*“ naredbu, primijetit ćemo da će se ovi uvjeti provjeriti točno jednom. Tada učenicima pomažemo uočiti da zbog toga što pomicemo glavnog lika po sceni, kameleonova pozicija će se mijenjati tijekom cijele igre. Zbog toga moramo stalno provjeravati te uvjete, ne samo jednom, nego doslovno cijelo vrijeme!

[Korak 5]

Za situacije kada moramo izvršavati određene naredbe tijekom cjelokupnog trajanja izvršavanja programa koristimo *zauvijek* petlju. Sve napisano ispod tijela *zauvijek* petlje izvršit će se iznova i iznova. S učenicima razgovaramo o tome da je u našem slučaju to upravo ono što želimo / trebamo kako bismo stvorili ovu igru.

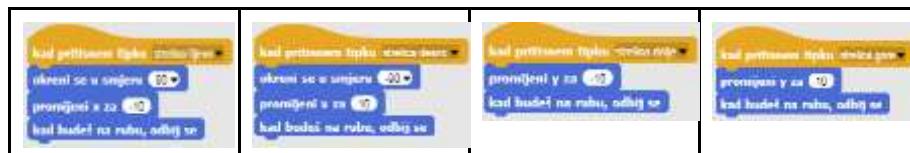
[Konačni kod]



[Prilagođavanje koda učenicima]

Da bismo pojednostavili ovu aktivnost, prethodno možemo pripremiti dio koda u datoteci predloška i uputiti učenike da ga dovrše.

Učenici koji su slijedili predloženi put učenja već su naučili kako pomicati objekt s tipkama. Dakle, možemo uključiti kôd kretanja u datoteku predloška. Učenici mogu promijeniti postavke tipki s tipki sa strelicama u neki vlastit raspored (npr. W, A, S, D).



Da bismo pomogli učenicima da razumiju koncept petlje **zauvijek** i kako je koristiti za otkrivanje boje pozadine, možemo uključiti kôd za otkrivanje dvije situacije: 1) objekt je u potpunosti u jednoj boji, 2) objekt dodiruje dvije boje istovremeno. Potrebno je uputiti učenike da popunjavaju kod za svaki od dva moguća slučaja.

Predloženi predložak koda:

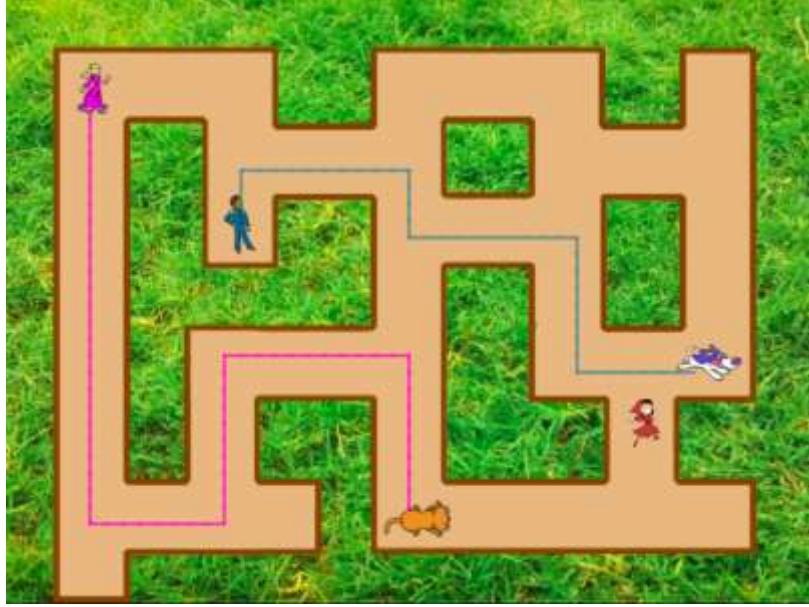


Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=zapusek&project=chameleon Lajović, S. (2011). Ogrebotina. Naučite kako programirati i postati računalna mačka. Ljubljana: Pasadena. Worderman, C. (2017). Računalo programiranje za djecu. Ljubljana: MK.	
Alati i materijali za učenike	Predložak izrađen u alatu Snap!: https://snap.berkeley.edu/project?user=zapusek&project=chameleon_template Polugotova igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=zapusek&project=chameleon_half_baked	

**Scenarij učenja 7 – Pomaganje princu i princezi u pronalasku njihovih životinja**

Naziv scenarija	Pomaganje princu i princezi u pronalasku njihovih životinja
Potrebno predznanje iz programiranja	Dodavanje teksta objektu Kretanje objekta pomoću strelica strelica koristeći događaje Korištenje uvjeta za <i>objekt dodiruje za stanje objekta</i> Korištenje događaja
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Uvjeti ako <i>objekt dodiruje određenu boju</i>● Kretanje po koordinatama● <i>Olovku digni, olovku pritisni</i>● Boja olovke Ishodi učenja: <ul style="list-style-type: none">● Učenik će moći koristiti naredbu <i>ako</i> za utvrđivanje stanja objekta i za vraćanje objekta natrag u slučaju da dotakne određenu boju● Učenik će moći postaviti objektu početne x i y koordinate● Učenik će moći koristiti <i>olovku digni</i> i <i>olovku pritisni</i> za crtanje linije/puta● Učenik će moći mijenjati boju olovke ovisno o paru kojeg spaja● Učenik će biti sposoban uvidjeti da na početku mora očistiti sve prethodno nacrtane putove
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Djevojčica mora pomoći princezi pronaći njezinu mačku i princu pronaći njegovog psa. To radi na način da otiđe do princeze i pokaže joj, crtanjem linije, put do njezine mačke; na sličan način djevojčica pokazuje princu put do njegovog psa. Na tom putu djevojčica mora izbjegavati susret životinja kako se putovi do njih ne bi preklapali. Zadatak: U prvom koraku učenici moraju odabratu prikladnu pozadinsku sliku (labyrinth). Dodaju u labirint pet objekata – vlastiti objekt (djevojčicu), princezu, princa, mačku i psa. Nadalje, programiraju kretanje djevojčice pomoću tipki (koristeći događaje), gdje moraju paziti da objekt ne hoda po travi. Nakon toga, programiraju crtanje olovkom i mijenjanje boje olovke koristeći događaje. Također, moraju isprogramirati početni događaj u kojem se raščisti put i djevojčica daje upute za početak igre. Cilj: Učenici će se upoznati s crtanjem pokretom tipke. Uz to će naučiti kako koristiti uvjete kojima će spriječiti da se objekt kreće po cijelome ekranu.
Trajanje	30 minuta



Strategija i metode učenja i poučavanja	Aktivno učenje Game-design Based Learning – igra Rješavanje problema
Oblici poučavanja	Frontalni rad Individualni rad
Razrada aktivnosti	<p>Učenicima je potrebno dati sljedeće materijale:</p> <ul style="list-style-type: none">• sliku pozadine• Snap! projekt u kojem je definiran objekt djevojčice• kôd za kretanje u jednom smjeru <p>Učenicima se prezentira problem: Djevojčica odluči pomoći princezi pronaći njezinu mačku i princu pronaći njegovog psa pokazivanjem (crtanjem) puta do njihovih životinja. Kako ne bi došlo do zabune, putovi trebaju biti različitih boja i ne smiju se sijeći.</p>  <p>[Korak 1] Učenici trebaju urediti pozadinsku sliku – labirint. Za uvođenje uvjeta “ako dodiruje boju” ili pozadina (trava) mora biti jednobojna ili put mora imati jednobojni okvir, kao u našem slučaju. Kako bi izbjegli navedene “probleme” s pronalaženjem prikladne pozadinske slike, dajemo im gore priloženu pozadinu.</p> <p>[Korak 2] Učenici na početku već imaju objekt djevojčice. Trebaju pronaći još četiri objekta i smjestiti ih u labirint. Za sve objekte trebaju namjestiti prikladne dimenzije (koje trebaju biti manje od širine putova u labirintu).</p> <p style="text-align: right;">postavi veličinu na 10 %</p>



[Korak 3]

Nakon toga moraju osmisliti kretanje djevojčice u četiri smjera korištenjem tipki. Pretpostavimo da to znaju napraviti iz prethodnih aktivnosti. Ipak, dajemo im kôd za jedan smjer koji im služi kao pomoć za izradu kôda za preostala tri smjera.



Kako bi kretanje djevojčice bilo što realnije (da se okreće lijevo i desno) kliknemo na *gledaj samo lijevo-desno* u bloku za smjer.



[Korak 4]

U idućem koraku učenici moraju spriječiti kretanje djevojčice po livadi. To rade tako da dodaju blok s uvjetom ako dodiruje smeđu boju. Ako djevojčica dodiruje smeđu boju (kraj puta), pomiče se 10 koraka unatrag. Ti koraci nam nisu vidljivi i čini se da djevojčica ostaje na istoj poziciji.



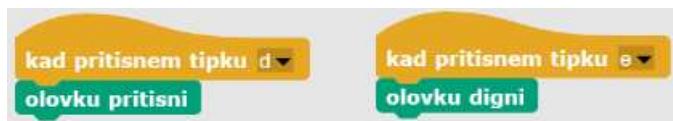
Učenici dodaju ovaj kôd ispod gore navedenog kôda za svaku strelicu, npr.:





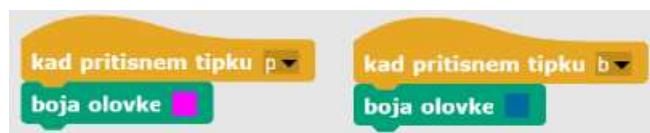
[Korak 5]

Nadalje, učenici programiraju crtanje s blokovima *olovku digni* i *olovku pritisni* korištenjem događaja *kad pritisnem tipku*.



Kada je tipka "D" pritisnuta i djevojčica se kreće, ona crta liniju.
Kada je tipka "E" pritisnuta, crtanje prestaje.

Na sličan način, učenici postavljaju boju olovke pritiskom tipke.



[Korak 6]

Na kraju, učenici programiraju događaj *kad kliknem na zelenu zastavicu*, gdje dodaju upute koje djevojčica na početku kaže. Nakon što odigraju igru nekoliko puta, učenici će vidjeti da je korisno dodati sljedeće blokove: *olovku digni* (za slučaj da ostane spuštena nakon prethodne igre), *obriši* (briše sve nacrtane putove iz prethodnih igara) i *kreći na x, y* (djevojčica uvijek započinje na tim koordinatama, koje su na putu i nisu na travi).

Kako bi odredili početne koordinate djevojčice, kliknemo na nju mišem i odvučemo ju na mjesto s kojeg želimo da započne igru. Nakon toga kliknemo na blok *Kretanje*, gdje možemo pronaći *položaj x* i *položaj y*. Klikom na *položaj x* saznajemo x koordinatu djevojčice, dok klikom na *položaj y* saznajemo y koordinatu.





	<p>[Cijeli kôd]</p> <p>[Dodatni zadatak] Učenici mogu upotpuniti igru dodatnim zadacima po želji ili koristiti sljedeće prijedloge:</p> <ul style="list-style-type: none">Postaviti početne coordinate za princa i princezu i napisati kod za njihovo kretanje. Postaviti odgovarajuću veličinu objekata. Crtati stazu do svojih životinja.Uključiti dodatne objekte.Svaki object treba crtati drugačijom bojom.Prilagoditi početne upute.Dodati upute igraču za pomicanje objekata (npr. tipkama W, S, A and D) i crtanje staze (npr. tipkom 3). <p>Alati i materijali za nastavnike Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=mateja&project=Helping%20Prince%20and%20Princess%20to%20find%20their%20animals Primjer igre s dodatnim zadacima: https://snap.berkeley.edu/project?user=mateja&project=Helping%20Prince%20and%20Princess%20to%20find%20their%20animals%20%2B%20Add.%20Task Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena. Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</p> <p>Alati i materijali za učenike Polugotova igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=mateja&project=Helping%20Prince%20and%20Princess%20to%20find%20their%20animals%20-%20Part</p>
--	--



Scenarij učenja 8 – Crtanje s kredom

Naziv scenarija	Crtanje s kredom
Potrebno predznanje iz programiranja	Dodavanje teksta objektu (sprite-u) Crtanje s olovkom (olovka gore, olovka dolje, postavljanje boje) Kretanje s koracima Petlje Događaji
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Petlja ponavljaj● Okretanje za 90 stupnjeva● Okretanje u smjeru● Mijenjanje pozadine Ishodi učenja: <ul style="list-style-type: none">● Učenik će koristiti petlju <i>ponavljaj</i> kad se isti blokovi ponavljaju 2-4 puta● Učenik će koristiti okretanje za 90 stupnjeva kod crtanja različitih oblika (kvadrat, pravokutnik, slovo „T“)● Učenik će objasniti značenje <i>okreni se u smjeru 90</i>● Učenik će moći promjeniti pozadinu pomoću događaja <i>kad pritisnem tipku</i>
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Igrač dobije tri različite pozadine i mora spojiti točke u tri različita oblika – kvadrat, pravokutnik i slovo „T“. Zadatak: Učenici odabiru pozadinu „pločaKvadrat“ i počinju crtati kvadrat. Početna pozicija je točka „A“. Kod crtanja kvadrata, ponavljaju određene korake 4 puta, stoga umjesto da pišu kôd 4 puta, mogu koristiti petlju <i>ponavljaj 4</i> . Nakon toga crtaju pravokutnik, također koristeći petlju <i>ponavljaj</i> . U ovom slučaju <i>ponavljaj 2</i> . U zadnjem zadatku trebaju spojiti točke u obliku slova „T“, te trebaju pronaći broj potrebnih koraka. Mogu koristiti petlju <i>ponavljaj</i> tako gdje je to moguće. Cilj: Učenici će se upoznati s crtanjem različitih oblika pomoću kôda. Naučit će koristiti petlju <i>ponavljaj</i> kako bi smanjili veličinu kôda i promjeniti pozadinu.
Trajanje	60 min
Strategija i metode učenja i poučavanja	Aktivno učenje Game Based Learning – igra Rješavanje problema
Oblici poučavanja	Frontalni rad Individualni rad /Rad u paru



Razrada aktivnosti

Za učenike su unaprijed pripremljene:

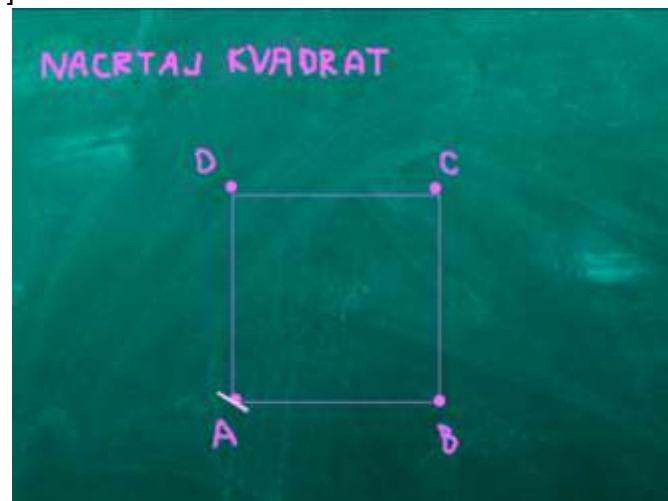
- tri pozadine sa svim točkama koje trebaju spojiti
- objekt „kreda“

Učenicima se prezentira problem:

Kreda želi nacrtati kvadrat, pravokutnik i spojiti točke u obliku slova „T“ ali ne zna kako se kretati i kako se okrenuti.

Napiši kôd i pokaži kredi kako da to radi!

[Korak 1]



Učenici počinju s ovom pozadinom. Pišu kôd za crtanje kvadrata. Počevši od točke „A“, kreće se X koraka do točke „B“, okrene za 90 stupnjeva lijevo, kreće X koraka do točke „C“, okrene za 90 stupnjeva lijevo, kreće X koraka do točke „D“, okrene za 90 stupnjeva lijevo, kreće X koraka do točke „A“ (i okrene za 90 stupnjeva lijevo).





Lakše je koristiti *okreni se 90 stupnjeva* budući da uvijek možemo koristiti okretanje za 90 stupnjeva (ovisi samo da li želimo da se okrene lijevi ili desno). Drugi način je koristiti *okreni se u smjeru 0, 90, 180, -90*, ali je malo komplikiranije jer moramo razdvojiti 4 mogućnosti i ne možemo koristiti petlju *ponavljam*. Blok *čekaj 1 s* je dodan zbog toga da vidimo crtanje (sve korake). Bez tog bloka, cijeli kôd se izvrši u sekundi. Učenici bi trebali isprobati napisati kôd bez tog bloka kako bi razumjeli značenje.

Pitamo učenike kako bi skratili kôd, ako je to moguće. Postoji li dio koji se može ponoviti? Odgovor je da postoji. Umjesto da piše isti kôd 4 puta, u programiranju koristiti petlju *ponavljam*.



Ako želimo vidjeti crtanje, moramo staviti blok *olovku pritisni* ispred petlje *ponavljam*.

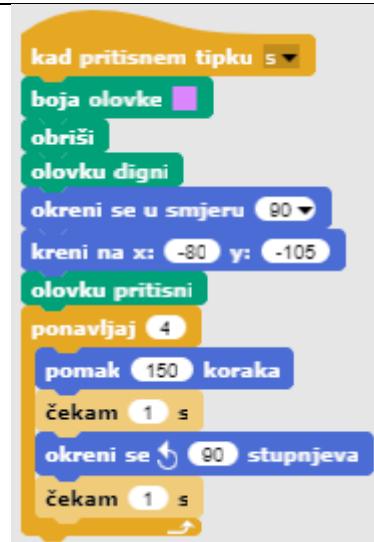


Ako ne želimo da se kreda rotira kod okretanja, u bloku za smjer odaberemo *ne rotiraj*.



[Korak 2]

Za pokretanje kôda, učenici koriste blok za događaj npr. *Kad pritisnem tipku s*. Također mogu postaviti *boju olovke*, i, kao što već znaju iz prethodnih aktivnosti, sljedeće blokove: *olovku digni* (u slučaju da je ostala dolje od prethodnog igranja), *obriši* (briše crtež prethodnog igranja) i *kreni na x,y* (da kreda uvijek krene s te pozicije). Ponekad se dogodi da zaustavimo program tijekom reprodukcije i da je objekt rotiran u „neobičnom smjeru“. To je problem kod ponovnog pokretanja igre. Ako je objekt rotiran pogrešno, ići će na primjer dolje, a ne desno u prvom koraku. Kako bi izbjegli taj problem, dodajemo blok *okreni se u smjeru 90*.



[Korak 3]

Nakon crtanja kvadrata, želimo nacrtati pravokutnik. Znači da moramo promijeniti pozadinu. To ćemo učiniti u dva koraka:

- Kliknemo na pozadinu (nazvanu *ploča*, s desne strane ekrana)

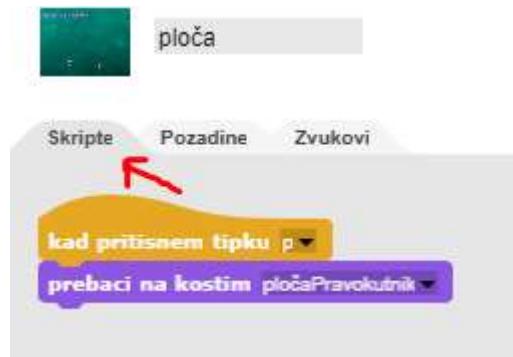


Klikom na *Pozadine* možemo vidjeti sve tri potrebne pozadine (*pločaKvadrat*, *pločaPravokutnik*, *pločaT*) spremne za ovu aktivnost.

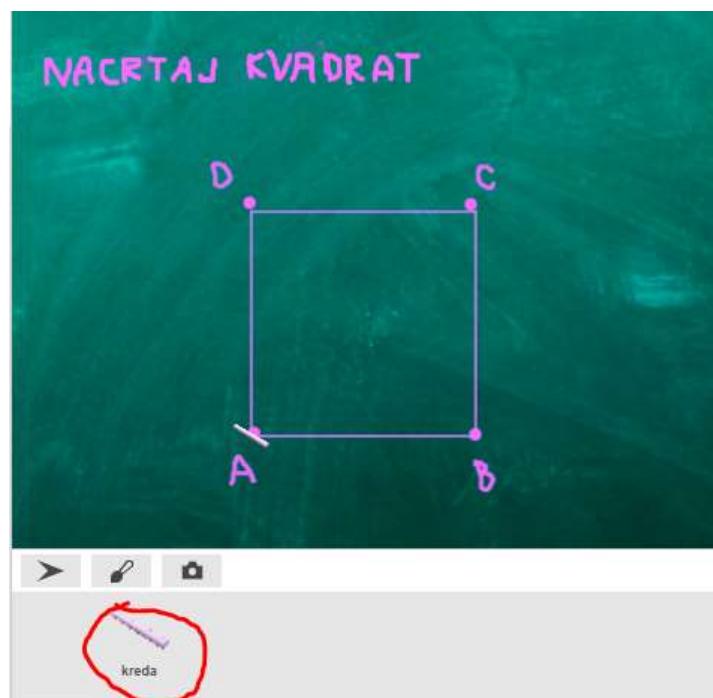




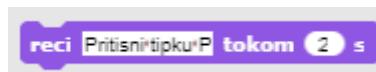
Za pisanje kôda, učenici trebaju kliknuti na gumb *Skripte*. Da bi program izmijenio pozadinu koristimo blok s događajem *kad pritisnem tipku p* i prebací na kostim pločaPravokutnik.



b) Kliknemo natrag na kredu.

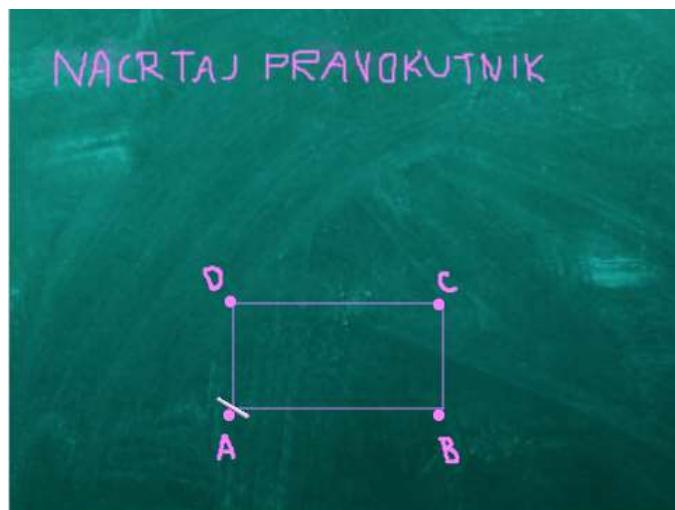


Ispod kôda iz [Korak 2] učenici dodaju blok, u kojem kažu igraču što da učini kako bi se pozadina promjenila, tj. da pritisne tipku „P“.

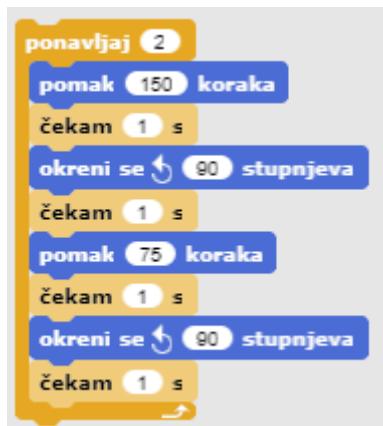




[Korak 4]



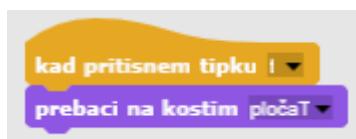
Nakon pritiska na tipku „P“, pozadina se mijenja u ovu na slici. Slično kao i prije, treba spojiti točke i nacrtati pravokutnik. Učenici mogu kopirati kôd za kvadrat i izmijeniti ga kako bi program nacrtao pravokutnik. Trebaju promijeniti petlju *ponavljam*. U ovom slučaju se petlja ponavlja 2 puta.



[Korak 5]

Nakon crtanja pravokutnika, učenici će spojiti točke u oblik slova „T“. To znači da će promijeniti pozadinu. U ovom koraku zapravo ponavljaju [Korak 3] i mijenjaju slovo „P“ u slovo „T“ i stavljaju kostim (PločaT).

- Kliknu na pozadinu (naziva *ploča*, s desne strane zaslona), gdje napišu kôd za promjenu pozadine. To će učiniti sa blokovima *kad pritisnem tipku!* i *prebac na kostim pločaT*.

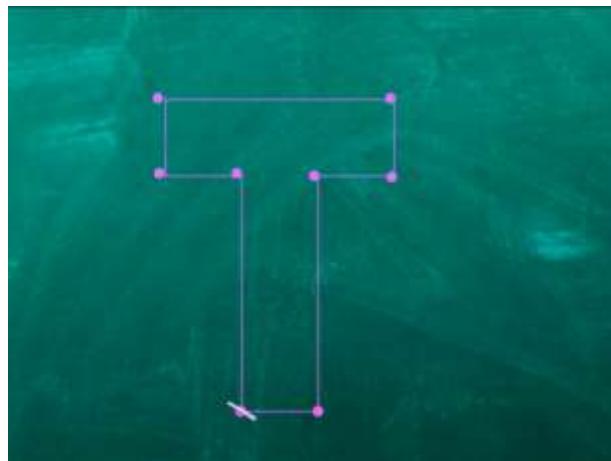


- Kliknu natrag na kredu i ispod kôda iz [Korak 4] dodaju blok,



u kojem kažu igraču što da učini kako bi se promijenila pozadina tj. da pritisne tipku „T“.

[Korak 6]



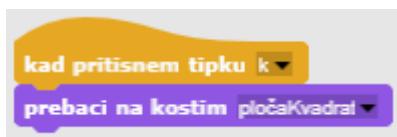
Nakon pritiskanja tipke „T“, pozadina se promjeni u ovu sliku. Slično kao i prije, trebaju spojiti točke i nacrtati slovo „T“. Učenici mogu kopirati kôd za kvadrat i izmijeniti ga. Učenici trebaju promijeniti početne koordinate, koje nisu jednake kao i prije. Iz prethodnih aktivnosti znaju kako odrediti točne koordinate. Nakon toga napišu kôd za crtanje slova „T“. Učenici trebaju odrediti broj koraka. Jedno od mogućih rješenja je:



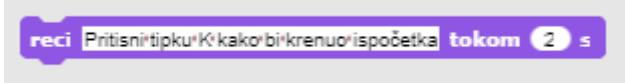
[Korak 7]

Nakon što smo promijenili pozadinu, ne možemo se vratiti na prvu pozadinu da bi nacrtali kvadrat. Stoga učenici trebaju dodati još jedan dio kôda. Ponavljam [Korak 3/5].

- Kliknu na pozadinu (naziva *ploča*, na desnoj strani zaslona), gdje napišu kôd za promjenu pozadine. To će učiniti s blokovima *kad pritisnem tipku k i prebac na kostim pločaKvadrat*.



- Kliknu natrag na kredu i ispod kôda iz [Korak 6] dodaju blok, u kojem kažu igraču što treba učiniti kako bi se promijenila pozadina, tj. da pritisne tipku „K“.





	<p>[Konačan kôd]</p>
Alati i materijali za nastavnike	<p>Učenici mogu uključiti u igru dodatne zadatke po želji ili koristeći sljedeće prijedloge:</p> <ul style="list-style-type: none">• Dodati novu pozadinu i nacrtati neke točke.• Napisati kod kojim se povezuje točke (koristeći novu ili postojeću pozadinu).
Alati i materijali za učenike	<p>Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=mateja&project=Drawing%20with%20a%20chalk</p> <p>Lajovic, S. (2011). Scratch. <i>Nauči se programirati in postani računalniški maček</i>. Ljubljana: Pasadena.</p> <p>Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i>. Ljubljana: MK.</p> <p>Polugotova igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=mateja&project=Drawing%20with%20a%20chalk%20-%20Part</p>

**Scenarij učenja 9 – Skupljanje otpadaka i čišćenje parka**

Naziv scenarija	Skupljanje otpadaka i čišćenje parka
Potrebno predznanje iz programiranja	Postavljanje početnih koordinata Postavljanje veličine objekta (sprite-a) Dodavanje teksta objektu Kretanje objekta pomoću strelica koristeći događaje
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">• Varijable• Prikazivanje i sakrivanje objekta• Kopiranje objekta• Kopiranje bloka kôda• Uvjeti Ishodi učenja: <ul style="list-style-type: none">• Učenik će koristiti varijable za brojanje pokupljenih otpadaka• Učenik koristiti sakrivanje objekta kada se on dotakne i prikazuje objekta na početku• Učenik može kopirati objekta (npr. od jedne boce napraviti 4 boce)• Učenik može kopirati blok kôda (iz objekta boce u objekt papira)• Učenik može koristiti uvjete za provjeru je li objekt prikidan i jesu li svi otpaci pokupljeni
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Park je pun otpadaka i djevojčica je odlučila počistiti ga. Kada pokupi sve otpatke, mora ih baciti u kantu za smeće. Zadatak: Učenici započinju s postavljanjem početnih koordinata lika djevojčice. Igra završava kada djevojčica pokupi sve otpatke i baci ih u kantu za smeće. Kako bi to napravili, učenici moraju koristiti varijable za prebrojavanje bodova (1 pokupljeni otpadak = 1 bod). Kada djevojčica dotakne otpadak, pokupi ga, otpadak se sakriva i broj bodova se poveća za 1. Kada djevojčica pokupi sve otpatke, odlazi do kante za smeće. Ako ode do kante za smeće prije nego pokupi sve otpatke, dobiva od kante poruku da se vrati kada pokupi sve otpatke. Cilj: Učenici će naučiti kako koristiti varijable i kako kopirati blok kôda ili čak cijeli objekt.
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje Rješavanje problema Game Based Learning – izrada igre
Oblici poučavanja	Frontalno poučavanje Individualni rad



Razrada aktivnosti

Učenicima je potrebno dati sljedeće materijale:

- sliku pozadine
- Snap! projekt u kojem je definiran objekt djevojčice (s kôdom za kretanje), objekt boce, objekt papira i objekt kante za smeće

Učenicima se prezentira problem: Djevojčica se želi prošetati i uživati u parku. Kada stigne u park, vidi da je park pun otpadaka. Odluči pokupiti sve otpatke. Kada to učini, napokon može leći na travu i uživati u čistom parku.



[Korak 1]

Postavljena je slika pozadine kao i objekt djevojčice s kôdom za kretanje pomoću strelica i uvjetom za dodir smeđe crte.



Učenici trebaju postaviti početne koordinate djevojčice koristeći blok *kreni na x, y*. Koordinate izaberu učenici, jedino je bitno da se djevojčica nalazi na stazi.



Također moraju dodati i uputu igraču, npr.:



[Korak 2]

Za prebrojavanje broja pokupljenih otpadaka, koristit ćemo varijable. Što je varijabla? Varijabla se može zamisliti kao kutija u koju spremamo informacije.

U našem slučaju, varijablu možemo zamisliti kao kutiju koja se zove bodovi. Kada djevojčica pokupi otpadak, on se spremava u varijablu *bodovi*. Ova varijabla broji koliko otpadaka je djevojčica pokupila. Kako stvorimo varijablu?



Odaberemo narančasti blok *Variable*, zatim kliknemo na gumb *Napravi varijablu*, u skočnom prozoru *Ime varijable* upišemo kako će se zvati naša varijabla i kliknemo OK. Pojavljuje se blok *bodovi*.

✓ **bodovi**

Nakon što se napravi, varijabla će biti prikazana na sceni. Ako ne želimo da se varijabla prikazuje na sceni, može se isključiti kvačicu kraj bloka *variable*.



Na početku igre, vrijednost varijable mora biti 0 jer još nije pokupljen niti jedan otpadak. Ispod kôda iz [Korak 1] učenik dodaje blok *postavi ___ na 0*. Klikom na padajući izbornik odabire odgovarajuću varijablu, koja je u ovom slučaju *bodovi*.



[Korak 3]

Učenici trebaju napisati kôd za objekt boce. Ideja je da objekt nestane (odnosno sakrije se) kada dotakne djevojčicu. Dakle, kôd će započeti kada objekt dodiruje djevojčicu.

Zatim je potrebno razmisiliti u kojem slučaju djevojčica pokupi otpatke. Ako kažemo da se otpadak sakrije kada je pokupljen,



možemo ga kupiti samo ako je i dalje na pozornici → prikazuje se. Ako je objekt (boca) i dalje na pozornici, kupimo ga i „stavimo u kutiju variabile“. Prije smo imali 0 elemenata u varijabli bodovi, sada imamo 1. Možemo vidjeti da sakupljanjem otpadaka mijenjamo vrijednost varijable (*bodovi*) za 1. Kada je otpadak kupljen, sakrijemo ga.



Sada možemo provjeriti je li kôd ispravan.

Kliknemo na zelenu zastavicu i kupimo bocu. Boca mora nestati i moramo imati 1 bod. Ako želimo ponovno igrati igru, ponovno kliknemo na zelenu zastavicu. Što se dogodi? Gdje je sada boca? Boca je sakrivena jer smo je prije sakupili. Dakle, moramo napraviti da se boca prikazuje na početku igre. To učinimo tako da odaberemo blok *pokaži*.



[Korak 4]

Ako želimo imati više boca u igri, kopiramo objekt boce: desni klik na objekt i odaberite duplicitiraj.



Mišem kliknite na novu bocu i odvucite je negdje unutar labirinta. Postupak duplicitiranja se ponovi tako da se dobiju 4 boce.

[Korak 5]

Sada je potrebno napraviti kôd za objekt papira. Moguće je duplicitirati kôd za bocu tako da se napravi desni klik mišem na blok kôda i odvuče na objekt papira tako da se mišem klikne na objekt *papir*.





Ovaj korak se ponavlja kako bi se duplicirao blok kôda kad kliknem na zelenu zastavu – pokaži.

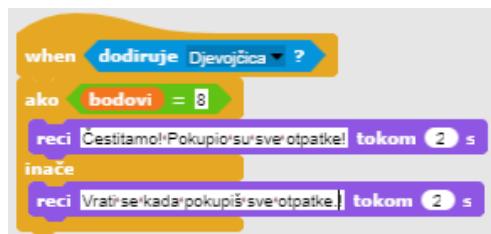
Također se ponovi i [Korak 4] te se duplicira cijeli objekt papira kako bi bilo više otpadaka papira u labirintu.

[Korak 6]

Posljednje što učenici trebaju napraviti je napisati kôd za kantu za smeće. Objekt za kantu za smeće je već napravljen, učenici ga mogu postaviti bilo gdje unutar labirinta.

Ovaj kôd će se također aktivirati kada djevojčica dotakne kantu za smeće.

Kanta za smeće treba provjeriti ako su svi otpaci pokupljeni. Pošto imamo varijablu bodovi, ovo će biti jednostavno za napraviti. Recimo da imamo 8 komada otpadaka u igri, potrebno je provjeriti ako je broj bodova jednak 8. Ako je, svi otpaci su pokupljeni. Ako je broj bodova manji od 8, djevojčica nije pokupila sve otpatke. Koristiti ćemo *ako-inache* izjavu kako bi to definirali u programu i dodat ćemo tekst igraču s informacijom je li pokupio sve otpatke ili nije.





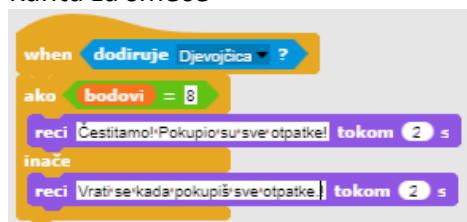
[Cijeli kôd]
Djevojčica



Boce / Papiri



Kanta za smeće



[Dodatni zadatak]

Učenici mogu uključiti u igru dodatne zadatke po želji ili koristeći sljedeće prijedloge:

- Dodati novu vrstu otpada (npr. bio otpad).
- Dodati što će reći kanta za smeće (npr. "Pokupili ste X boca, Y papira i Z lubenica").
- Ako igrač pokupi svo smeće, kanta za smeće može čestitati



	<p>igraču.</p> <ul style="list-style-type: none">• Ako igrač ne pokupi svo smeće, kanta za smeće ga može upozoriti na to te navesti koje smeće je ostalo.
Alati i materijali za nastavnike	<p>Igra izrađena alatom Snap!:</p> <p>https://snap.berkeley.edu/project?user=mateja&project=Picking%20up%20trash%20and%20cleaning%20the%20park</p> <p>Primjer igre s dodatnim zadacima:</p> <p>https://snap.berkeley.edu/project?user=mateja&project=Picking%20up%20trash%20and%20cleaning%20the%20park%20%2B%20Add.%20Task</p> <p>Lajovic, S. (2011). <i>Scratch. Nauči se programirati in postani računalniški maček</i>. Ljubljana: Pasadena.</p> <p>Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i>. Ljubljana: MK.</p>
Alati i materijali za učenike	<p>Polugotova igra izrađena alatom Snap!:</p> <p>https://snap.berkeley.edu/project?user=mateja&project=Picking%20up%20trash%20and%20cleaning%20the%20park%20-%20Part</p>



Scenarij učenja 10 – Hranjenje mačaka

Naziv scenarija	Hranjenje mačaka
Potrebno predznanje iz programiranja	Postavljanje pozadine Postavljanje objekata Dodavanje teksta objektu Korištenje varijabli
Ishodi učenja	Koncepti programiranja <ul style="list-style-type: none">• Varijable• Petlje• Slučajni brojevi• Spajanje stringova• Operatori: logički, aritmetički• Unos <p>Ishodi učenja:</p> <ul style="list-style-type: none">• Učenik će moći prepoznati situaciju u kojoj će koristiti petlju za ponavljanje n puta• Učenik će moći razlikovati dodjeljivanje vrijednosti u svakoj iteraciji petlje i prije same petlje• Učenik će moći koristiti blok unos da bi dobio broj od „igrača“• Učenik će moći koristiti aritmetičke operatore za generiranje pravog odgovora• Učenik će moći koristiti naredbu ako-onda za provjeru ispravnosti unosa i daje odgovarajuće odgovore• Učenik će moći koristiti varijablu koja će brojati točne odgovore
Cilj, zadaci i kratki opis aktivnosti	Kratak opis: Programirajte igru u kojoj će igrač morati izvršiti deset izračuna množenja i računati odnosno brojati točne odgovore. Zadatak: Programirajte aktivnost u kojoj će Marta, čuvarica skloništa, neprestano pitati igrača za broj mačaka koje mora nahraniti u određenoj sobi. Broj mačaka ovisi o broju i veličini zdjela. Navedene veličine se za svaku sobu moraju dodijeliti nasumično. Također, moramo imati brojač koji će brojati točne odgovore. Prvo, čuvarica skloništa mora objasniti zadatak za igrače te onda započinje igra. Igra završava kada ona zatraži broj mačaka 10 puta. Svaki puta mora odgovoriti je li broj koji se upiše točan ili nije. Nakon aktivnosti mora dati sažetak u kojem će napisati koliko je igrač bio uspješan odnosno reći će koliko je puta igrač odgovorio ispravno odnosno koliko je puta pogriješio. Cilj: Učenici će biti upoznati sa konceptom dodjele slučajne vrijednosti varijabli unutar petlje te će razlikovati kada navedeno napravimo van petlje. Naučit će kako dobiti, testirati i izbrojati ispravne unose igrača.



Trajanje	45 min
Strategija i metode učenja i poučavanja	Aktivno učenje Suradničko učenje Rješavanje problema Game Based Learning-izrada igre
Oblici poučavanja	Frontalno poučavanje Rad u paru/Individualni rad/Grupni rad
Razrada aktivnosti	<p>Učenicima se prezentira problem: Čuvarica skloništa pokušava nahranići svoje mačke u 10 različitih soba. U svakoj sobi je slučajni broj zdjela (2-10) različitih veličina (1-5), ali unutar pojedine sobe sve su zdjele jednake veličine. Veličina zdjele nam govori koliko mačaka može jesti iz iste, npr. ako je veličina zdjele 3 to znači da iz nje mogu jesti 3 mačke. Pomozite pronaći broj mačaka koje mora nahranići u svakoj sobi.</p> <p>[KORAK 1]</p> <p>Prvo upućujemo učenike da izaberu zanimljivu pozadinu za igru. Ako želimo uštedjeti na vremenu možemo im odmah ponuditi pozadinu.</p>  <p>[KORAK 2]</p> <p>Zatim, moramo odabrati novi objekt koji će predstavljati čuvara mačjeg skloništa.</p>  <p>[KORAK 3]</p> <p>Za pohranu potrebnih vrijednosti potrebne su nam 3 varijable: 1) za spremanje broja točnih odgovora, 2) za dodjelu slučajne vrijednosti za broj zdjela unutar svake sobe (2-10) i 3) za dodjelu slučajne vrijednosti veličine zdjela (1-5). Brojač točnih odgovora morati će biti postavljen na 0 dok za preostala dva brojača nećemo trebati dodijeliti točnu vrijednost prije petlje. Razlog tome je što ćemo im u svakoj iteraciji petlje dodijeliti novu slučajnu vrijednost. Također,</p>

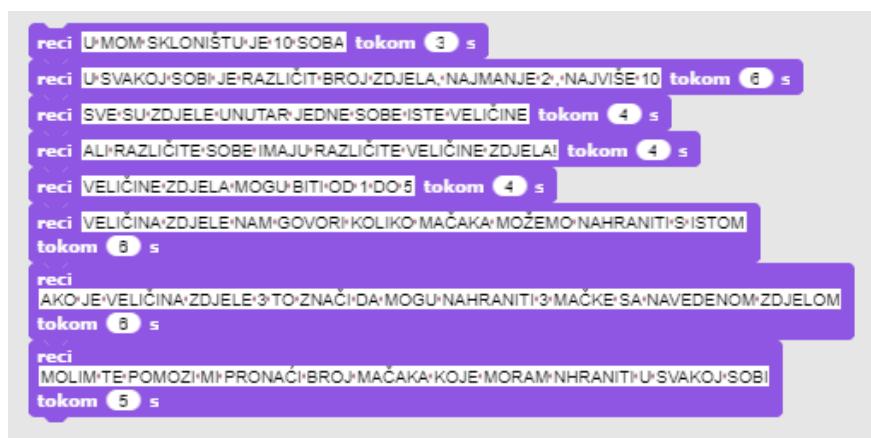


želimo brojati sobe, ali nam ne treba posebna varijabla za to. Upotrijebiti ćemo istu varijablu kao i u petlji, postaviti ćemo ju na 1 a zatim će se povećavati u svakoj iteraciji za 1 sve dok taj broj ne poprimi vrijednost 10. Navedena petlja također služi i za brojanje soba.



[KORAK 4]

Zatim, moramo programirati upute za igrače. To radimo pomoću Izgled (reci) i čekaj određeni broj sekundi (n).

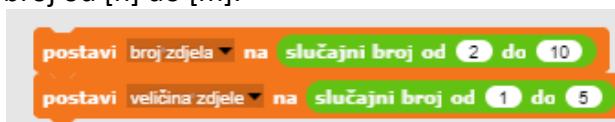


[KORAK 5]

Razgovaramo sa učenicima o tome koje radnje će se dogoditi u svakoj sobi i hoće li biti iste. Odnosno koje će se naredbe koristiti tj. smjestiti u petlju. Prvo ćemo morati nasumično odrediti vrijednost (1-10) za broj zdjela i za veličinu zdjele (1-5). Nakon toga morati ćemo pitati igrača koliko mačaka možemo nahraniti u pojedinoj sobi. Nadalje, morati ćemo ispitati točnost odgovora i dati odgovarajući odgovor (povratnu informaciju) i zapamtiti ga ako je točan (brojati točne odgovore). Na kraju svake iteracije (ponavljanja) morati ćemo povećati broj soba za 1.

[KORAK 6]

Za dodjeljivanje slučajne vrijednosti za broj zdjela i njihovu veličinu upotrijebit ćemo Variable/ postavi [opcija] vrijednost na Operatori / slučajni broj od [n] do [m].



[KORAK 7]

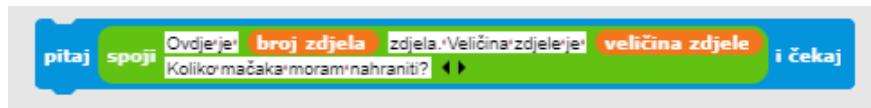


Želimo pitati igrače za broj mačaka koje je potrebno nahraniti i to koristeći Osjetila/ pitaj [string] i čekaj, jer će se u protivnom prikazati nekoliko sekundi, a zatim ažurirati novim retkom teksta. Na taj način igrači mogu lako zaboraviti koliko zdjela/veličina je u trenutnoj sobi. Kako bi napravili rečenicu koja se sastoji od kombinacije teksta i referenci na varijable koristimo blok Operatori/spoji [string1][string2]. Navedeni blok morat će proširiti tako da stane cijela rečenica.



[KORAK 8]

Cijelu prethodnu rečenicu moramo staviti unutar bloka Osjetila/ pitaj [string] i čekati da bi dobili odgovor igrača.



[KORAK 9]

Nakon što igrač odgovori moramo provjeriti ispravnost odgovora. Moguće su dvije situacije, igrač može odgovoriti točno ili netočno, pa ćemo koristiti naredbu ako-onda. Točan odgovor je vrijednost umnoška broja zdjela i veličine zdjele. Moramo provjeriti je li odgovor igrača jednak tom broju. Ako je odgovor točan, povećavamo brojač točnih odgovora za 1 i dajemo odgovor. Ako je odgovor netočan dajemo samo odgovor. Ne moramo brojati netočne odgovore jer ih možemo izračunati uz pomoć brojača točnih odgovora.



[KORAK 11]

Sada moramo odabrati neku petlju. Kao što je već rečeno najbolje je koristiti petlju jer varijabla koju koristimo za ponavljanje replicira prebrojavanje soba.

[KORAK 12]



	<p>Kada se petlja zaustavi, igra je gotova. Na kraju dajemo informacije o postignuću igrača. Broj točnih odgovora spremljen je u brojaču točnih odgovora; broj netočnih odgovora možemo naknadno izračunati.</p> <p>[Cijeli kod]</p> <pre>when green flag clicked set [pokusaji v] set [macak v] say [Hey! Jaz sam Marta, čuvanica skloništa mačaka] for (2) seconds ask [Kako ti je ime?] and wait answer = answer say [Imam jednu igricu za tebe!] for (3) seconds say [Možeš li pogoditi u 5 pokušaja koliko mačaka imam u svom skloništu?] for (4) seconds say [Otkrij čuti nešto!] for (2) seconds say [Uvijek imam barem jednu i najviše 100 mačaka] for (5) seconds repeat [ponavljam dok ne bude odgovor = broj macaka] if [odgovor < macak v] then say [Što misliš, koliko mačaka imam?] and wait change [pokusaji v] by (-1) if [odgovor > macak v] then say [Ne, ne... Imam više mačaka!] and wait change [pokusaji v] by (-1) else say [Imam manje mačaka!] and wait end if [pokusaji v < 0] then say [Super! Pogodio si točan broj mačaka!] and wait if [broj pokusaja < 6] then say [Pogodio si rezultat u 5 pokušaja ili čak i manje i zato ćeš dobiti vlastitu mačku] and wait end end end if [macak v > 5] then say [Inače] and wait say [Premašio si broj pokušaja za dobiti vlastitu mačku. Pokušaj ponovno zaigrati.] and wait end</pre>
Alati i materijali za nastavnike	<p>Cijela aktivnost u Snap!-u:</p> <p>https://snap.berkeley.edu/project?user=zapusek&project=cat_feeding_2</p> <p>Lajovic, S. (2011). Scratch. <i>Nauči se programirati in postani računalniški maček</i>. Ljubljana: Pasadena.</p> <p>Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i>. Ljubljana: MK.</p>
Alati i materijali za učenike	<p>Predložak izrađen u Snap!:</p> <p>https://snap.berkeley.edu/project?user=zapusek&project=cat_feeding_template</p>



CODING4GIRLS
2018-1-SI01-KA201-047013



Co-funded by the
Erasmus+ Programme
of the European Union

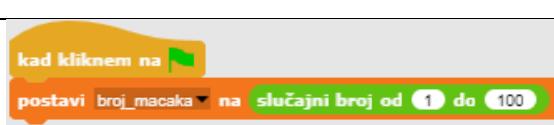
--	--

**Scenarij učenja 11 – Pogađanje broja mačaka u skloništu**

Naziv scenarija	Pogađanje broja mačaka u skloništu
Potrebno predznanje iz programiranja	Dodavanje pozadine. Dodavanje novog objekta. Definirati što objekt govori.
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">• Nasumične vrijednosti• Variable• Unos vrijednosti od strane korisnika• Petlja <i>ponavljam dok</i>• Uvjeti• Operatori uspoređivanja• Brojač Ishodi učenja: <ul style="list-style-type: none">• Učenik će pohraniti nasumičnu vrijednost u varijablu• Učenik će koristiti blok za unos podataka kako bi pohranio pokušaj pogađanja broja mačaka• Učenik će koristiti uvjete (<i>naredbu ako</i>) i operatore uspoređivanja za provjeru vrijednosti variable i ispis odgovarajućeg teksta• Učenik će koristiti petlju <i>ponavljam dok</i> kako bi igraču omogućio više pokušaja pogađanja i provjerio upisanu vrijednost• Učenik će implementirati brojač u petlji za prebrojavanje broja pokušaja pogađanja igrača• Učenik će postaviti uvjet za završetak izvođenja petlje da bi definirao kraj igre
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Djevojčica Marta volontira u skloništu za mačke. Igrač mora pogoditi broj mačaka koji se trenutno nalazi u skloništu Zadatak: Sklonište za mačke u kojem djevojčica volontira uvijek ima između 1 i 100 mačaka. Na početku se zadaje nasumičan broj mačaka koji igrač mora pogoditi. Marta pita igrača da pogodi trenutni broj mačaka. Nakon što igrač unese broj, djevojčica daje jedan od odgovora: 1) ako je upisani broj manji od stvarnog broja, kaže: "broj mačaka je veći", 2) ako je upisani broj veći od stvarnog broja, kaže: "broj mačaka je manji", 3) ako je upisani broj točan, kaže: "Odlično, pogodio si točan broj!". Igrač može pogađati 5 puta. Ako pogodi, igrač dobiva mačku, u suprotnom je ne dobiva. Cilj: Učenici će se upoznati s petljom <i>ponavljam dok</i> i kako postaviti uvjet koji zaustavlja igru. Također će naučiti kako koristiti variable u različitim situacijama: za pohranjivanje nasumične vrijednosti, kao brojač ili za pohranjivanje vrijednosti koju upiše igrač.
Trajanje	45 minuta

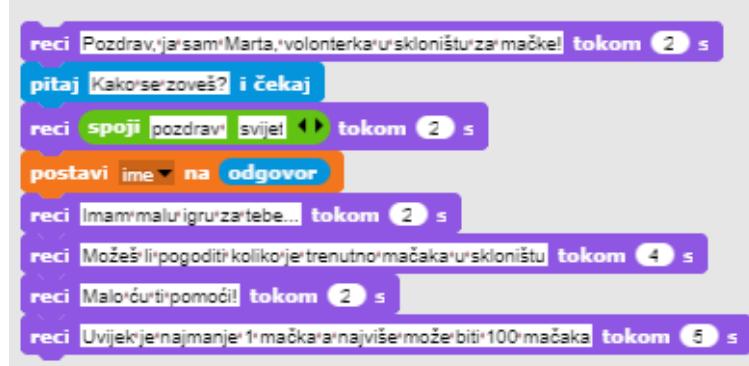


Strategija i metode učenja i poučavanja	Aktivno učenje Rješavanje problema Game Based Learning – izrada igre
Oblici poučavanja	Frontalno poučavanje Individualni rad / rad u paru / grupni rad
Razrada aktivnosti	<p>Učenicima se prezentira problem: Djevojčica Marta, volonterka u skloništu za mačke, želi da pogodiš točan broj mačaka u skloništu. Broj može biti između 1 i 100. Kada igrač upiše broj, Marta odgovori je li broj veći, manji ili je to točan broj mačaka. Ako igrač pogodi broj u manje od 5 pokušaja, dobiva mačku. U suprotnom, djevojčica zatraži da se igra ponovno.</p> <p>[Korak 1]</p> <p>Prvi zadatak je napraviti zanimljivu pozadinu igre. Učenici mogu sami nacrtati ili koristiti slobodne slike s interneta. Kako bi se uštedjelo na vremenu, pozadina je unaprijed pripremljena.</p>  <p>[Korak 2]</p> <p>Potrebno je odabrat novi kostim za zadani objekt koji će biti volonterka u skloništu.</p>  <p>[Korak 3]</p> <p>Raspravljamo s učenicima kako je ova igra zanimljiva ukoliko se igra više puta pošto se broj mačaka postavlja nasumično. Kako bi postavili nasumičan broj, potrebno ga je spremiti u varijablu. Varijabla je sada jedini način da se zapamti određena vrijednost u Snap!-u (prepostavljamo da još uvijek nisu upoznati s konceptom liste). To se mora dogoditi kada se igra pokrene (Upravljanje → Kada kliknem na zelenu zastavu).</p>



[Korak 4]

Djevojčica treba pitati igrača njegovo ime kako bi ga pozdravila. To je moguće napraviti koristeći blok *pitaj i čekaj* koji se nalazi unutar skupine *Osjetila*. Igračev odgovor se spremi u već definiranu varijablu *odgovor* (nalazi se u skupini *Osjetila*). Kako bi Marta pozdravila igrača, potrebno je povezati ime koje se spremilo u varijablu *odgovor* s pozdravom. To se napravi pomoću bloka *spoji[rijec1][rijec2]* koji se nalazi u skupini *Operatori*. Za prikaz teksta koristi se blok *reci tokom 2 sekunde* iz skupine *Izgled*. Isti blok se koristi i za pisanje uputa igraču. Učenicima je potrebno naglasiti da trebaju biti pažljivi oko vremena koliko se tekst prikazuje.



[Korak 5]

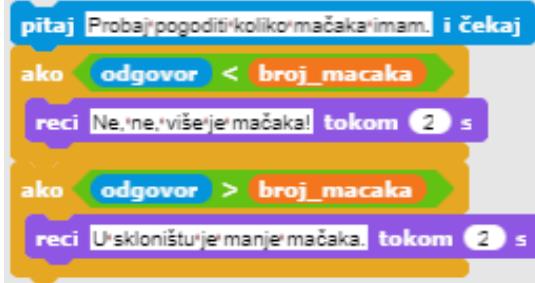
S učenicima se diskutira kako nije moguće predvidjeti koliko puta će igrač pogodati kako bi došao do točnog broja. Igrač može pogoditi jako brzo i pogoditi u prvom pokušaju, možda će mu biti potrebno 5 pokušaja ili čak i više. Iz tog razloga je potrebno odabrati pravu petlju za ovaj zadatak.

Volonterka treba uzastopno pitati i dati odgovarajući odgovor sve dok igrač ne pogodi točan broj. Jedina petlja koju možemo koristiti je petlja *ponavljam dok ne bude*. Uvjet je relativno jednostavan - moramo ponavljati dok odgovor igrača, koji je spremjen u varijabli *odgovor*, nije jednak vrijednosti koja je spremljena u varijabli *broj_mačaka*.



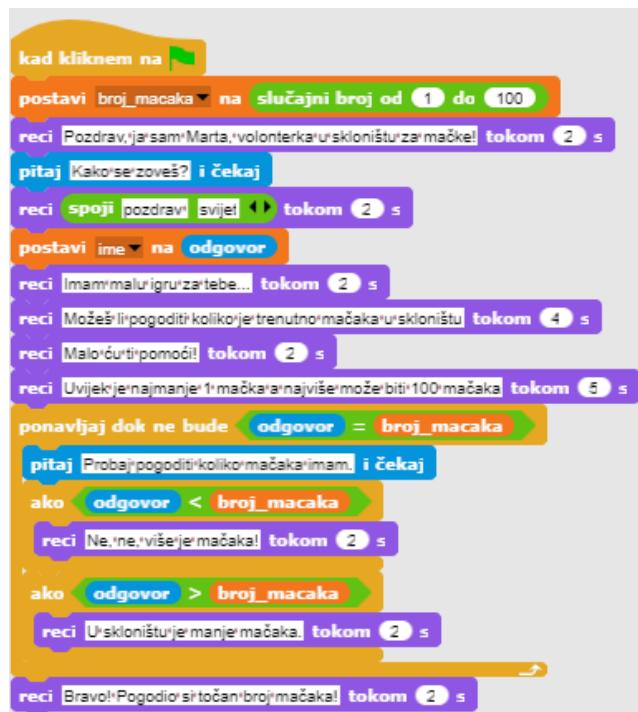
[Korak 6]

Zatim, moramo pitati učenike koje naredbe moramo uključiti u tijelo petlje. Koja se aktivnost ili naredba ponavlja dok igrač ne pogodi točan broj? Prvo, potrebno je pitati igrača da upiše broj, zatim moramo dati odgovor ovisan o vrijednosti upisanog broja.



[Korak 7]

Posljednja stvar za objasniti ili diskutirati s učenicima je što uzrokuje kraj izvršavanja petlje. Kada je igračev odgovor jednak broju mačaka, oba uvjeta u tijelu petlje će biti pogrešna i petlja će otici u iduću iteraciju provjeravajući uvjet petlje. Ovog puta, uvjet će biti istinit, pa će se petlja prekinuti i izvest će se naredbe koje dolaze nakon petlje. Drugim riječima, kada se petlja završi znamo da je igrač pogodio točan broj i možemo ispisati poruku igraču u skladu s time.



[Korak 9]

Kako je broj pokušaja za pogađanje ograničen, moramo napraviti novu varijablu koja će imati ulogu brojača i postaviti početnu vrijednost na 0. S učenicima diskutiramo o važnosti inicijalizacije varijable i razlike između postavljanja vrijednosti i povećanja vrijednosti (kada postavimo vrijednost varijable, prethodna vrijednost se izgubi). Svaki put kada igrač upše broj želimo brojač povećati za 1.



[Korak 10]

Nakon točnog odgovora, moramo provjeriti vrijednost varijable brojača kako bi odlučili hoće li igrač dobiti mačku ili ne. Pošto Snap! ima samo logički operator manje (<), a nema operator manje ili jednako, uvjet je je li *brojač_mačke* < 6. Ovo je također dobar primjer korištenja *ako-inače* uvjeta jer razlikujemo dva slučaja.

[Cijeli kôd]

The screenshot shows a Scratch script titled "cats in a shelter". It starts with a "when green flag clicked" hat block. Inside, there's a "set [broj_pokusaja v.] to [0]" control block. A "repeat (5)" control block follows, containing a "say [Pozdrav, ja sam Marta, volonterka u skloništu za mačke] for [2] seconds" and a "say [svijet] for [2] seconds" block. The repeat loop ends with a "set [broj_macaka v.] to [random (1) to (100)]" control block. Next is a "say [Imam malu igru za tebe...]" for 2 seconds, followed by a "say [Možeš li pogoditi koliko je trenutno mačaka u skloništu?]" for 4 seconds, and a "say [Malo duriš pomoći!]" for 2 seconds. Then comes a "say [Uvijek je najmanje 1 mačka a najviše može biti 100 mačaka]" for 5 seconds. A "repeat (10)" control block begins with a "say [Probaj pogoditi koliko mačaka imam.]" for 2 seconds, followed by an "ask [odgovor]" and a "if then" branch. If the answer is less than the current value of "broj_macaka", it says "Ne, ne, više je mačaka!" for 2 seconds. If the answer is greater, it says "U skloništu je manje mačaka." for 2 seconds. After the repeat loop ends, a "say [Bravo! Pogodio si točan broj mačaka!]" for 2 seconds, followed by an "if then" branch for "broj_pokusaja < 6". If true, it says "Dobio si mačku jer si pogodio i manje od 6 pokušaja" for 2 seconds. Otherwise, it says "Imao si previše pokušaja. Ali uvijek možešigrati ponovno!" for 2 seconds.

Alati i materijali za nastavnike

Igra izrađena alatom Snap!:

https://snap.berkeley.edu/project?user=zapusek&project=cats_in_a_shelter

Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.

Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.

Alati i materijali za učenike

Predložak izrađen u alatu Snap!:

https://snap.berkeley.edu/project?user=zapusek&project=cats_in_a_shelter_template

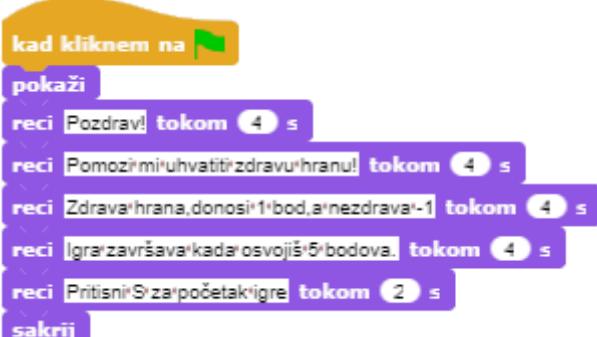


NAPREDNI SCENARIJI UČENJA

Scenarij učenja 12 – Hvatanje zdrave hrane

Naziv scenarija	Hvatanje zdrave hrane
Potrebno predznanje iz programiranja	Dodavanje teksta objektu Prikazivanje i skrivanje objekta Korištenje točke za određivanje smjera Korištenje nasumičnih vrijednosti Korištenje varijabli za brojenje bodova Korištenje petlje <i>ponavljam</i> Korištenje petlje <i>zauvijek</i> Korištenje uvjeta
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Varijable● Uvjeti● Petlje● Točka za smjer● Nasumične vrijednosti Ishodi učenja: <ul style="list-style-type: none">● Učenik će moći koristiti varijablu za sprječavanje pokretanja igre prije nego djevojka završi s govorom (izborno)● Učenik će moći koristiti uvjet <i>ako</i> kako bi provjerio (uz pomoć variabile) može li se hrana početi kretati● Učenik će moći koristiti petlju ponavljam za kretanje hrane sve dok je broj bodova manji od 5● Učenik će moći koristiti točku u smjeru 180 (prema dolje) za objekte koji se kreću prema dolje● Učenik će moći koristiti nasumične vrijednosti za određivanje broja koraka● Učenik će moći koristiti nasumične vrijednosti za pomak na nasumičnu poziciju● Učenik će moći koristiti nasumične vrijednosti za pomak na poziciju x (nasumična vrijednost), y (fiksna vrijednost) (izborno)
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Djevojka hvata hranu. Mora biti oprezna, samo zdrave namirnice donose bodove! Zadatak: Učenici trebaju programirati dva različita objekta, djevojku koja daje upute, govoriti što treba učiniti za početak igre i broj bodove; i hranu koja nasumično pada s vrha zaslona. Uz to, učenici mogu dodati varijablu i uvjet <i>ako</i> za sprječavanje kretanja hrane sve dok djevojčica ne prestane govoriti. Cilj: Učenici će naučiti kako nasumično pomicati za X koraka i odabrati položaj i također kako koristiti varijable i uvjete za sprječavanje drugih događaja.



Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje Game Based Learning – izrada igre Rješavanje problema
Oblici poučavanja	Individualni rad / Rad u paru
Razrada aktivnosti	<p>Učenicima se prezentira problem: Djevojka treba uhvatiti hranu. Svaka zdrava namirnica donosi 1 bod, dok svaka nezdrava oduzima 1 bod. Igra počinje s nekoliko uputa koje daje djevojčica. Tada ona nestaje i pojavljuje se hrana. Kad igrač sakupi 5 bodova, hrana nestaje i djevojka se ponovo pojavljuje.</p>  <p>[Korak 1] Ova aktivnost namijenjena je za individualni rad ili rad u paru. Učitelj daje sugestije, objašnjava teže dijelove i pomaže kada je potrebno. Učenici biraju pozadinu i dodaju glavni lik, npr. djevojku. Djevojka na početku daje upute, a zatim nestaje. Kao što smo vidjeli u prethodnim aktivnostima, dobro je napisati blok <i>prikaži kad kliknem zastavu</i> (kada se ponovo igra, ako se objekt ne pokaže). Kod glasi, na primjer:</p>  <pre>kad kliknem na pokaži reci Pozdrav! tokom 4 s reci Pomozi mi uhvatiti zdravu hranu! tokom 4 s reci Zdrava hrana donosi 1 bod, a nezdrava -1 tokom 4 s reci Igra završava kada osvojiš 5 bodova. tokom 4 s reci Pritisni S za početak igre tokom 2 s sakrij</pre> <p>Na ovaj objekt ćemo se vratiti kasnije. Sad napišemo kod za voće.</p>



[Korak 2]

Učenici dodaju novi objekt, zdravu hranu, npr. jabuku.

Najprije, programiraju pokret, odozgo prema dolje, pa odabiru sljedeće blokove:



Ako ne žele da njihova jabuka bude naopako, učenici mogu odabrati treću opciju *ne okreći se u* bloku smjera.



Da bi igra bila zanimljivija, broj koraka može biti nasumično odabran, tako da brzina ne bude uvijek ista. na primjer:



Sljedeći korak je razmišljanje o tome što se događa kad jabuka dođe do dna zaslona?

U ovom slučaju učenici mogu koristiti blok *dodiruje rub* u kombinaciji s *uvjetom ako*. Ako jabuka dodirne rub, pomaknut će se na neki nasumični položaj. Blokovi za kretanje nude nam sljedeći blok:



Ova naredba će nasumično odabrati bilo koje x i y koordinate i jabuka se može pojaviti bilo gdje na zaslonu (pogledajte crvene točke na slici).



Ako želimo da se jabuka uvijek pojavi na vrhu zaslona, vrijednost y se može fiksirati, a samo x vrijednost se odabire nasumično.



Sljedećim kodom jabuka će se uvijek pojaviti na vrhu zaslona (pogledajte crvene točke na slici).

kreni na x: slučajni broj od -200 do 200 y: 150

[Korak 3]

Studenti sada mogu kreirati varijablu, *bodovi*, koju će koristiti za brojanje. Bodovi na početku moraju biti postavljeni na 0 (na sprite-u djevojke).

postavi bodovi ▾ na 0

[Korak 4]

Ako želimo da se jabuka pomiče po nekom obrascu, potrebna nam je petlja. Učenici mogu koristiti petlju *ponavljam dok* i postaviti uvjet. Na primjer, žele da igra završi kada osvoje 5 bodova. Dakle, uvjet će biti *bodovi = 5* i petlja će se ponavljati dok je uvjet lažan. Kad je uvjet istinit, ako igrač dosegne 5 bodova, petlja će se zaustaviti.

ako 5 = bodovi

[Korak 5]

Ne želimo da se jabuka prikaže na početku, nego nakon što djevojčica da upute. Učenici mogu programirati jabuku tako da se pokaže kad se pritisne tipku. Naravno, učenici trebaju dodati blok *pokaži* prije petlje ponavljam i blok sakrij se nakon toga. Cijeli kod zasada izgleda ovako:

pokaži
ponavljam dok ne bude 5 = bodovi
okreni se u smjeru 180
pomak slučajni broj od 1 do 2 koraka
ako dodiruje rub ?
kreni na x: slučajni broj od -200 do 200 y: 150
sakrij



[Korak 6]

Što se događa kad se klikne na jabuku (ili se unese miš)?

Jabuka se mora nestati, pribrojiti bodove, promijeniti položaj i ponovo se pokazati. Bodovi će se mijenjati za 1, a za poziciju učenici mogu koristiti isti kod kao i prije.

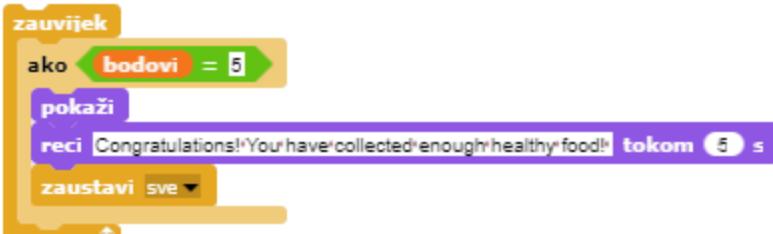


[Korak 7]

Vratimo se djevojci.

Djevojka se sada mora ponovo pojaviti i reći, npr. Čestitam!

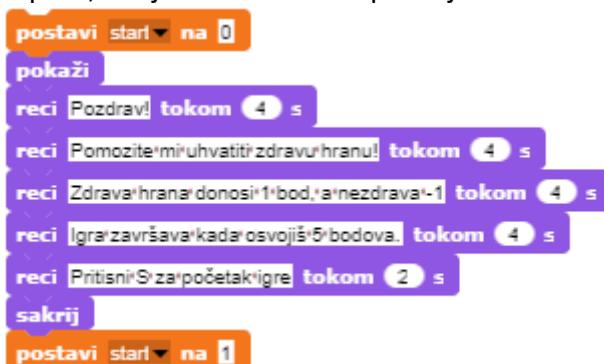
Trebat će nam trebati petlja *zauvijek* koja će provjeravati jesmo li osvojili 5 bodova. Ako smo osvojili 5 bodova, djevojka će pokazati i reći nešto. Nakon toga dodat ćemo blok *zaustavi sve*. Neka učenici sami dokuče što znači taj blok (bez zaustavljanja, djevojka će zauvijek govoriti "Čestitam ...").



[Korak 8]

Igrajući igru ponovo, kada učenici već znaju sve upute (od [Korak 1]) sigurno će ih htjeti preskočiti. Učenici mogu pritisnuti tipku S prije nego što igra počne, ali djevojka će još uvijek govoriti.

Da bismo to spriječili, možemo stvoriti drugu varijablu (nazvanu *start*), koju na početku moramo postaviti na 0. Tada, nakon djevojčinih uputa, varijabla *start* će se promijeniti u 1.



Sada moramo programirati jabuku da se pokrene samo ako je



varijabla *start* jednaka 1, što će učenici učiniti uvjetom if. Uz to, učenici neće moći pokrenuti igru prije nego što djevojčica prestane govoriti.

Kod jabuke je sada:



[Korak 9]

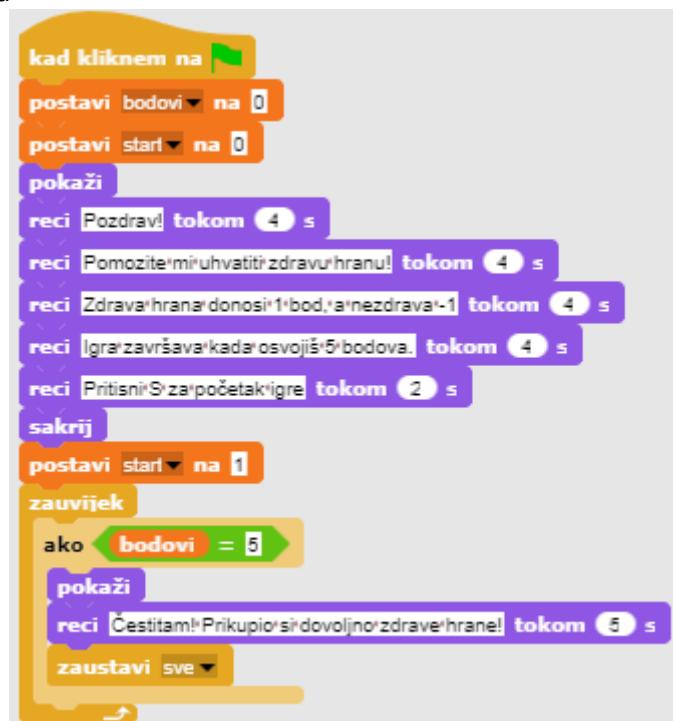
Studenti sada mogu duplicirati objekt jabuke i mijenjati mu kostim (ako žele). Kod će biti isti.

Jedina promjena je s nezdravom hranom, gdje će učenici klikom na nju izgubiti jedan bod.

promijeni varijablu *bodovi* za -1

[Konačni kod]

Djevojka



Jabuka



[Dodatni zadatak]

Učenici mogu dopuniti igru po želji ili koristiti sljedeće prijedloge:

- Dodati novi object – zdjelu (koristeći sliku koju ćete pronaći online ili koristeći priloženu sliku zdjele).
- Promijeniti igru tako da objekt zdjela hvata hranu.
- Postaviti početnu pozivi zdjele i napisati kod za njeno pomicanje.
- Promijeniti pravila – neka igra završava kada igrač prikupi 20 bodova (igrač pobjeđuje) ili kada pokupi 3 nezdrave namirnice (igrač gubi).
- Dodati još objekata koji predstavljaju hranu kako bi igra bila zanimljivija.
- Promijeniti kostim zdjele kada igrač prikupi 5, 10 ili 15 bodova.

Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=mateja&project=Catching%20healthy%20food Primjer igre s dodatnim zadacima: https://snap.berkeley.edu/project?user=mateja&project=Catching%20healthy%20food%20%2B%20Add.%20Task Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena. Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Alati i materijali za učenike	Polugotova igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=mateja&project=C4G12_Catching%20healthy%20food%20-%20Part



Scenarij učenja 13 – Pričam ti priču

Naziv scenarija	Pričam ti priču
Potrebno predznanje iz programiranja	Prikazivanje i sakrivanje objekata Uvjeti Upotreba <i>reci</i> Upotreba <i>pričekaj ... sekundi</i>
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Pomicanje i promjena veličine● Prikazivanje poruka i pričanje● Sastavljanje strukture priče● Mijenjanje pozadine prizora Ishodi učenja: <ul style="list-style-type: none">● Učenici će moći planirati dijaloge i aktivnosti u priči● Učenici će moći emitirati poruke i prikazivati govor (dijalog)● Učenici će moći pomicati i mijenjati veličinu objekata● Učenici će moći prikazivati i sakrivati objekte● Učenici će moći proširiti kod objekata
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Zec priča priču o Alisi u zemlji čudesa. Započinje s par rečenica, a u pozadini je prizor Alise. Priča započinje u šumi. Alisa šeta i pita se „Gdje sam?“ (Da bismo shvatili kako se Alise kreće, postupno se njezina veličina smanjuje). Alisa dolazi do raskrižja i ugleda mačku na drvetu. Započinje razgovor između Alise i mačke. Zadatak: Učenici moraju eksperimentirati s kratkim primjerom priče o susretu Alise i mačke na temelju sinkronizacije dijaloga koristeći blok za čekanje. Nadalje, pregledavaju drugu verziju priče pomoću prikazanih poruka. Upisuju se naredbe za razmjenu poruka. Učenici dovršavaju kôd, prema tekstu sa slike. Zadatak postaje kompleksniji mijenjanjem scenske pozadine i pomicanjem Alise kroz šumu prije njezinog susreta s mačkom. Cilj: Učenici će naučiti kako ispričati priču, kako koristiti poruke i kako promjeniti pozadinu pozornice.
Trajanje	90 minuta
Strategija i metode učenja i poučavanja	Game Based Learning – igra Aktivno učenje Rješavanje problema
Oblici poučavanja	Frontalni rad Rad u paru Individualni rad



Razrada aktivnosti

1. Učitelj razgovara sa učenicima o Alisi u zemlji čудesa i pokazuje sliku Alise koja se susrela s mačkom. Objašnjava da se Alisina priča može ispričati uz alat Snap!. Učenici imaju zadatak pokrenuti projekt i pogledati kodove objekata.
- https://snap.berkeley.edu/project?user=ddureva&project=Alice_1

S učenicima se vodi rasprava oko sljedećih pitanja:

- Tko prvi počinje razgovarati?
- Kada se Alisa uključuje u razgovor, a kada Mačka?
- Zašto u dijalogu likova nema sinkronizacije?

Odgovor leži u netočnom proračunu vremena u kojem se prikazuju poruke likova koji razgovaraju i nedostatku vremena da lik završi svoje odgovore.

The screenshot shows two Scratch scripts side-by-side. The top script is for character 'Alisa' and the bottom one is for 'Mačka'. Both scripts have a green flag sensor at the top.

Alisa's Script:

- When green flag clicked:**
 - Move x: -187 y: -67
 - Wait 9 seconds
 - Say [Pozdrav! for 2 seconds]
 - Say [Možete li mi molim vas reći kojim putem bih trebala krenuti? for 10 seconds]

Mačka's Script:

- When green flag clicked:**
 - Move x: -74 y: 113
 - Show [Cat] face
 - Wait 10 seconds
 - Say [Ovisi kamo želiš idti. for 10 seconds]

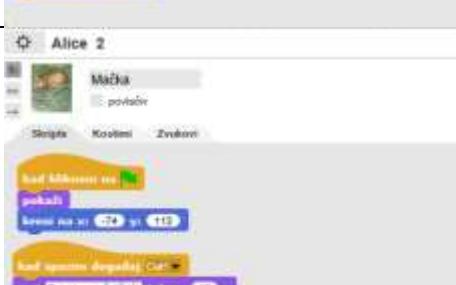


Objekt	Aktivnost	Početak	Završetak	Trajanje
Zec	Reci: Pozdrav! Jesi li čuo za Alisu i njezine avanture u zemlji čудesa? Pogledajmo njezinu priču.	0	14	14
Alisa	Reci: Možete li mi molim vas reći kojim putem bih trebala krenuti?	9	21	12
Mačka	Reci: Ovisi kamo želiš ići.	10	20	10

Zaključak je da sinkronizacija s blokom *pričekaj...* i drugim blokom može dovesti do pogreške u ponašanju likova koji razgovaraju.

2. Učitelj pokreće i pregledava se projekt https://snap.berkeley.edu/project?user=ddureva&project=Alice_2 te se postavlja pitanje koje su dosad nepoznate naredbe?

Nadalje, uspoređuju se kodovi dvaju projekata: Alice_1 i Alice_2:

Alice_1	Alice_2
	
	
	



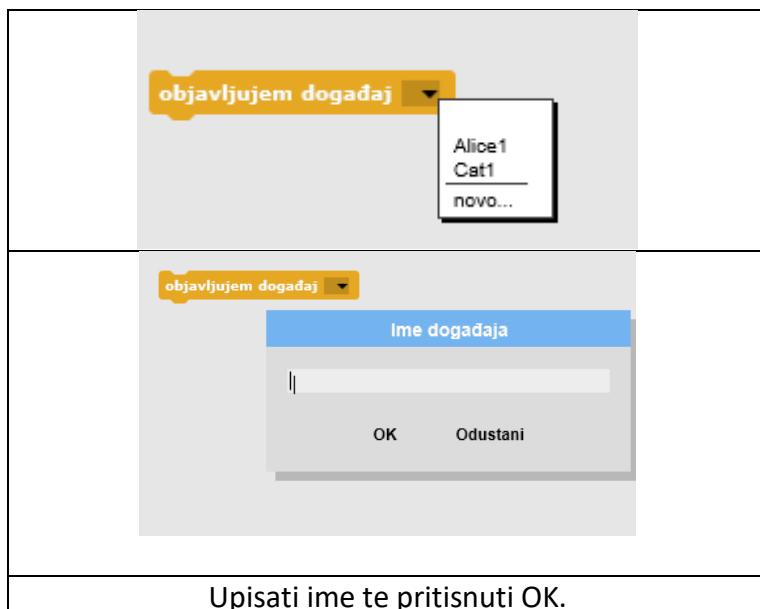
Uvode se blokovi za emitiranje poruka, odnosno događaja:

objavljujem događaj ▾ objavljujem događaj ▾ i čekam kad spazim događaj ▾

Raspravlja se o tome da emitirane poruke (događaji) šalju se svima, ali mogu ih primiti samo neki likovi.

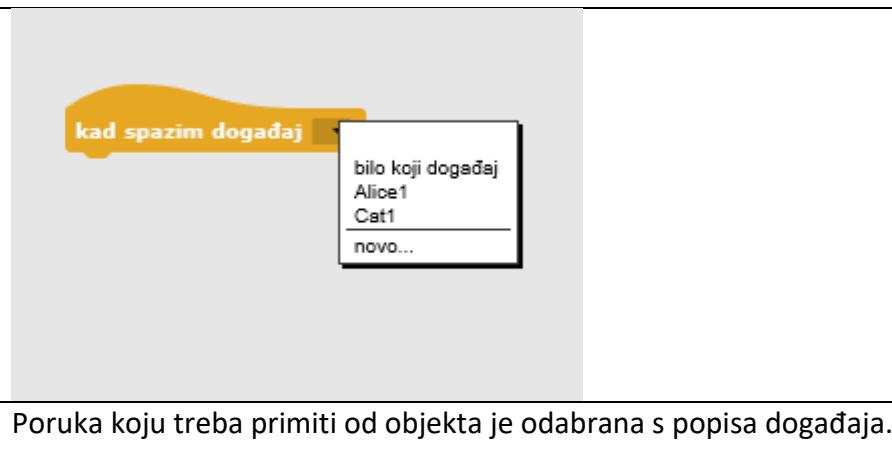
Objavljujem događaj ... i čekam zahtijeva da svi koji su primili poruku izvrše svoje radnje, a zatim se nastavljaju aktvnosti objekata.

Učitelj pokazuje kako imenovati događaj i kako se koristi u slučaju "Kad spazim događaj"...



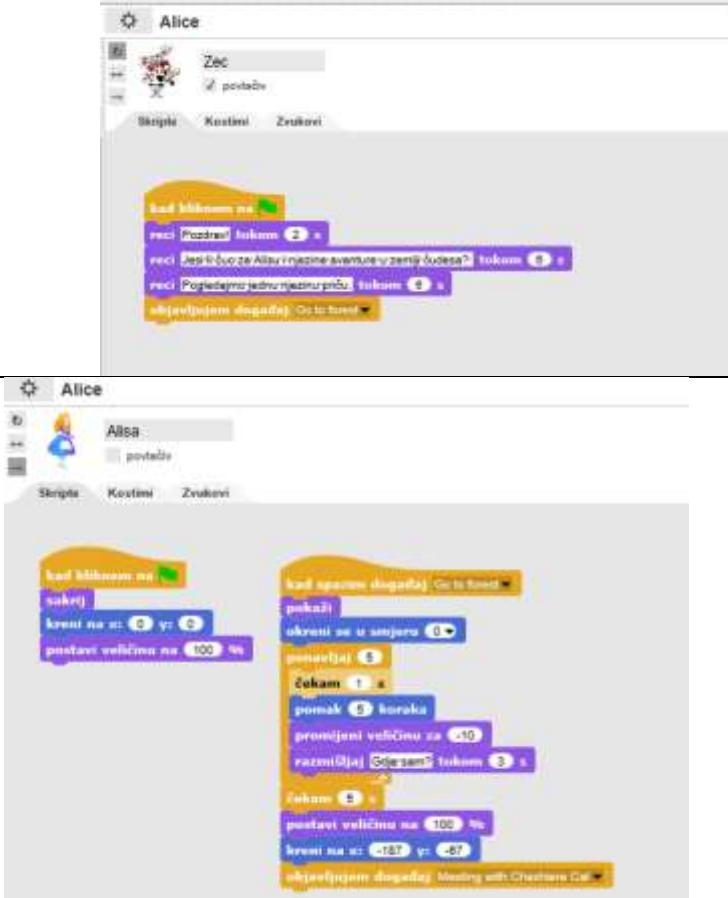
Upisati ime te pritisnuti OK.

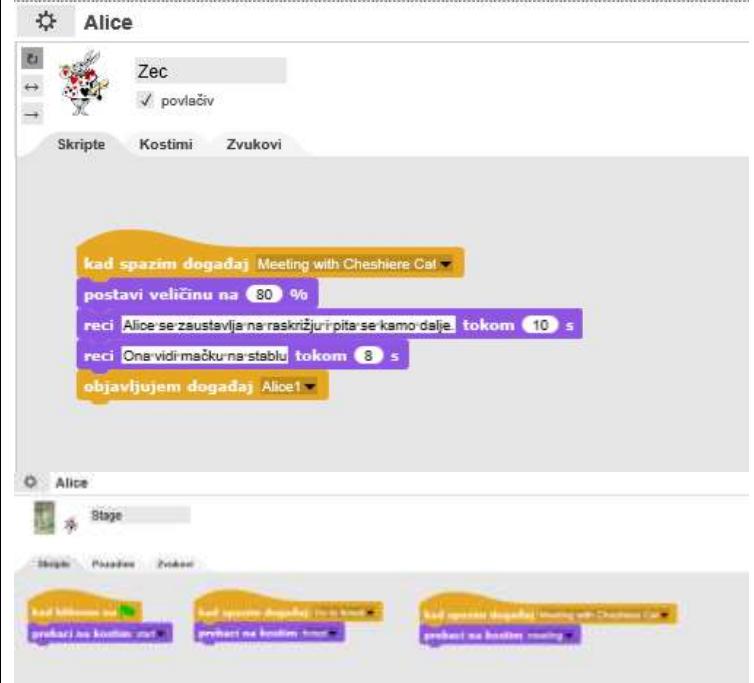
Upotreba u događaju (*Kad spazim događaj..*):



Poruka koju treba primiti od objekta je odabrana s popisa događaja.

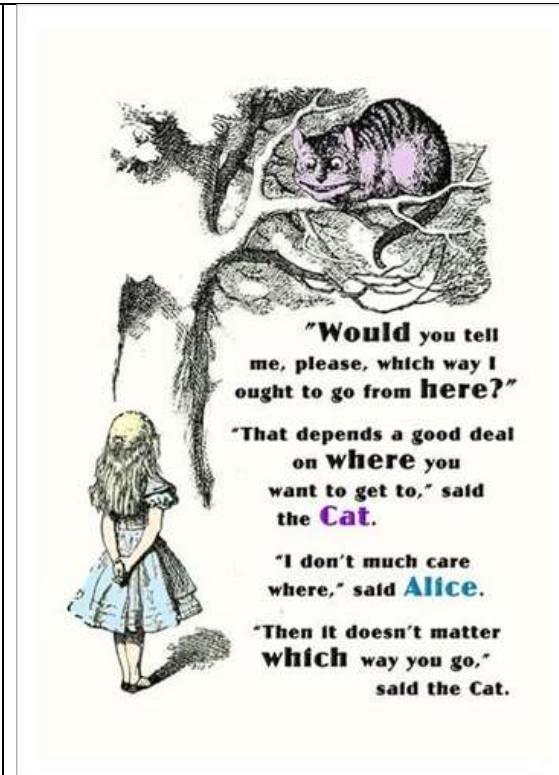


	<p>Raspravlja se kako dovršiti priču u slici te kako imenovati poruke.</p> <p>Učenici dovršavaju priču radom u paru.</p> <p>Učitelj naglašava da pričanje često zahtijeva promjenu scenskih pozadina. Cilj je napraviti priču cjelovitijom tako što će započeti priču sa Zecom na uvodnoj pozadini, premještajući radnju u šumu gdje Alisa hoda i pita se "Gdje sam?". Njena se veličina postupno smanjuje kako se odmiče sve dalje. Zatim se nađe na raskrižju i vidi mačku te započinje razgovor između njih dvoje.</p> <p>Učitelj prikazuje gotov projekt:</p> <p>https://snap.berkeley.edu/project?user=ddureva&project=Alice_2</p> <ul style="list-style-type: none">Komentiraju se promjene u scenama i radnjama likova. <i>"Kada se prizor mijenja? Kada se Alisa pojavljuje i koje su njene aktivnosti? Kada se mačka pojavljuje i koje su njene aktivnosti?"</i>Raspravlja se o scenama u projektu Alice_2. Postoje 3 scene, jedna je već korištena. <i>Koju scenu koristiti za početak? Što treba učiniti da se objekti Alisa i Mačka ne bi prikazali na početku projekta? Kako promjeniti scensku pozadinu?</i> 
--	---

	 <p>Dok je Alisa na putu u šumu, ona šeće, tj. udaljava se pa se njezina veličina smanjuje za -10%. To se ponavlja 5 puta pomoću petlje za ponavljanje.</p> <p>Kad stigne do raskrižja, prizor (pozadina) se mijenja s porukom "Susret s Mačkom". Ovu poruku istovremeno prima i Zec te se i njegova veličina smanjuje na 80% i nastavlja pričati sa smanjenom veličinom. U ovoj fazi, Mačka nije prikazana jer je prisutna kao dio ukrasa na drvetu. Pojavljuje se na Cat1 poruci. Učitelj može objasniti da je mačka izrezana iz ukrasa pomoću vanjskog grafičkog uređivača.</p> <p>Nakon objavljivanja poruke Zeca, priča se nastavlja.</p> <ol style="list-style-type: none"> 3. Učitelj komentira da, kako bi ispričala priča, prvo se mora izmislitи заплет. Za opis scenarija priče može se koristiti dodatna tablica. Učitelj odlučuje hoće li učenicima dati gotovu tablicu ili djelomično dovršenu te ju učenici vođeni ilustracijom, mogu dovršiti. 4. Učenici bi trebali opisati napravljeni scenarij te u paru dovršiti priču na projektu Alice_2.
Alati i materijali za nastavnike	Aktivnost izrađena alatom Snap! https://snap.berkeley.edu/project?user=ddureva&project=Alice



Alati i materijali za
učenike



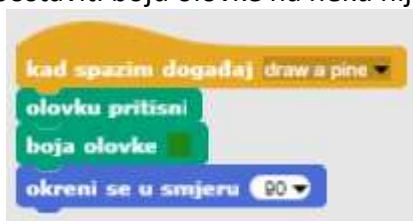
https://snap.berkeley.edu/project?user=ddureva&project=Alice_1

https://snap.berkeley.edu/project?user=ddureva&project=Alice_2

**Scenarij učenja 14 – Crtanje**

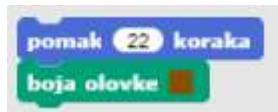
Naziv scenarija	Crtanje
Potrebno predznanje iz programiranja	Dodavanje objekta Korištenje pokazivača smjera Korištenje varijablu za brojanje bodova Korištenje naredbe ponavljanja (petlje) Korištenje uvjetovanja
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Varijable● Uvjetovanja● Petlje● Pokazivač smjera● Operatori Ishodi učenja: <ul style="list-style-type: none">● Učenici će moći koristiti opciju olovke za crtanje● Učenici će moći koristiti emitiranje za kontrolu objekta● Učenici će moći mijenjati vrijednost variable prilikom crtanja novih stabala● Učenici će moći koristiti petlje za crtanje dijelova stabala● Učenici će moći koristiti uvjete za promjenu pozornice● Učenici će moći koristiti pokazivač smjera (okreni se u smjeru) za crtanje krošnje stabala● Učenici će moći koristiti petlje ponavljanja za crtanje● Učenici će moći koristiti operator > za promjenu pozornice
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Klima se jako promjenila, zrak je radi industrije jako zagađen. Kako bi se poboljšala kvaliteta zraka potrebno je posaditi stabla! Zadaci: Kako bi se poboljšala kvaliteta zraka, učenici moraju programirati objekt pomoću kojeg će crtati dvije vrste različitih stabala – borove i hrast te gume koje simboliziraju te vrste stabala. Klikom na gumb, crta se određena vrsta stabla. Cilj: Učenici će naučiti kako se crta pomoću Snap!-a, promijeniti boju i deblijinu olovke te kako koristiti varijable i uvjete koji uzrokuju novi događaj.
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Game Based Learning – igra, aktivno učenje, rješavanje problema
Oblici poučavanja	Rad u paru / Individualni rad



Razrada aktivnosti	<p>Učenicima se prezentira kako treba izgledati igra: Na početku igre na pozornici se prikazuje industrija koja uzrokuje klimatske promjene i varijabla koja prikazuje kvalitetu zraka. Potrebno je posaditi stabla kako bi se poboljšala kvaliteta zraka. Crtaju se dvije različite vrste stabala, bor i hrast. Kada se nacrtava bor, zrak se poboljša za 3 boda, a crtanjem hrasta bor se poboljša za 2 boda. Kada kvaliteta zraka dođe na 10 bodova, pozadina se promjeni na livadu.</p> <p>[Korak 1]</p> <p>Ova aktivnost zamišljena je za individualan rad ili rad u paru. Učiteljica daje prijedloge, objašnjava teže dijelove i pomaže kada zatreba.</p> <p>Učenici za pozadinu postavljaju i prikazuju sliku industrije koja zagađuje zrak te dodaju objekt olovke. Pošto je objekt previelik, potrebno ga je smanjiti tako da se njegova veličina postavi na 50% i zadaje se početna pozicija olovke, npr. x=-10, y=-10. Primjer koda:</p>   <p>[Korak 2]</p> <p>Objekt olovke bi trebao primiti poruke „oak“ i „pine“ i crtati različita stabala kao odgovor na poruku. Označite objekt olovke i dodajte kod tako da kada objekt primi poruku „pine“ crta bor.</p> <p>Olovku je potrebno rotirati za 90 stupnjeva kako bi krošnja bila u obliku trokuta te postaviti boju olovke na neku nijansu zelene boje.</p>  <p>Za crtanje krošnje bora potrebno je pomaknuti objekt 40 koraka i nakon svakog koraka okrenuti se ulijevo za 120 stupnjeva.</p>  <p>Ovo kretanje je potrebno ponoviti 3 puta.</p>
---------------------------	---



Nakon što se završilo s crtanjem krošnje, treba nacrtati deblo. Objekt olovke se pomakne za 22 koraka i boja olovke se promijeni u smeđu.

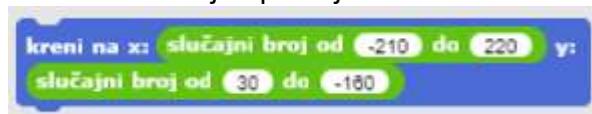


Olovka se okrene udesno za 90 stupnjeva i pomakne 10 koraka.



Sve se ponovi 3 puta.

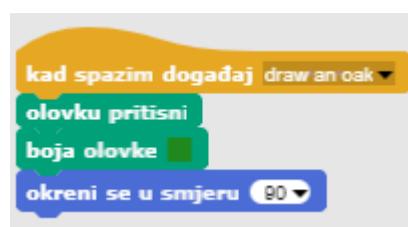
Kada se završi sa crtanjem potrebno je podignuti olovku kako ne bi napravila crtu prilikom pomicanja objekta olovke. Također, olovka se premješta na novu slučajnu poziciju.



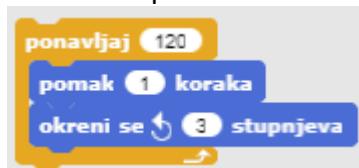
[Korak 3]

Na sličan način je potrebno omogućiti crtanje hrasta.

Označite objekt olovke i dodajte kod tako da kada objekt primi poruku „oak“ crta hrast. Kako bi krošnja bila u obliku kruga, olovka treba biti spuštena i zarotirana za 90 stupnjeva a njena boja zelena.

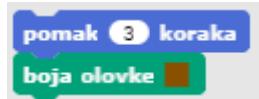


Za crtanje krošnje hrasta pomaknite se za 1 korak, okrenite ulijevo za 3 stupnja te to ponovite 120 puta.

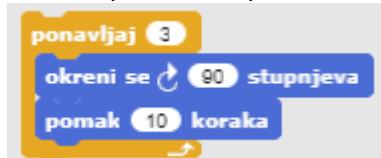




Nakon krošnje, treba nacrtati i deblo. Olovku je potrebno pomaknuti na sredinu krošnje za 3 koraka te postaviti smeđu boju.



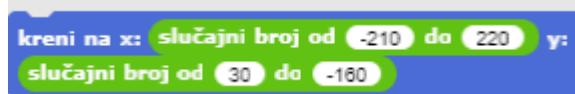
Zatim objekt okrenite 90 stupnjeva udesno te pomaknite za 10 koraka. Taj dio je potrebno ponoviti 3 puta.



Na kraju je potrebno podignuti olovku kako objekt ne bi ostavljao trag prilikom pomicanja.

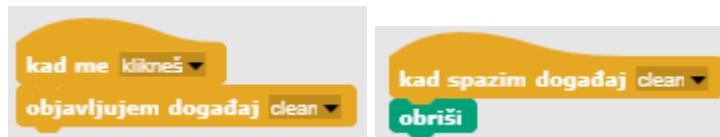


Nakon što je hrast nacrtan, olovku treba pomaknuti na novu slučajnu poziciju.



[Korak 4]

Nakon toga je potrebno omogućiti da se klikom na objekt za brisanje maknu sva nacrtana stabla. Kada se klikne na objekt „X“ šalje se poruka o brisanju. Nakon što primi tu poruku, olovka obruše drveće.



[Korak 5]

Napravite novu varijablu „clean air“ koja će pokazivati kvalitetu zraka. Na pozornici postavite da je na početku igre je vrijednost varijable jednaka 0.



Svaki put kad se nacrtava bor zrak se poboljša za 2 jedinice. Dodajte kod objektu olovka kojim će se povećati vrijednost varijable „clean air“ za 2 svaki put kada se klikne na bor.



Svaki put kad se nacrtava hrast zrak se poboljša za 3 jedinice. Dodajte



kod objektu olovka kojim će se povećati vrijednost varijable „clean air“ za 3 boda svaki puta kada se klikne na hrast.



[Korak 6]

Dodajte kod koji omogućuje da kada varijabla „clean air“ poprimi vrijednost 10, pozornica se promjeni u travu.

Na pozornici dodajte novu pozadinu „grass“ (iz materijala).



Označite objekt olovka i dodajte blok „When“ iz bloka „Upravljanje“.



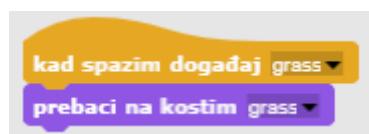
Iz bloka „Operatori“ dodajte operator >.



Napravite da objekt pošalje poruku „grass“ kada je varijabla „clean air“ veća od 10.



Označite pozornicu i dodajte kod da kada primi poruku „grass“ promjeni kostim na „grass“.



[Završni kod]
Bor (pine)



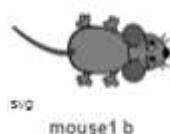
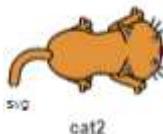
	 kad me klikneš▼ objavljujem događaj draw a pine▼
	Hrast (oak) kad me klikneš▼ objavljujem događaj draw an oak▼
	X kad me klikneš▼ objavljujem događaj clean▼
	Olovka kad kliknem na zeleni trapez postavi crtež - nap B olovka učišča olovko 3 kreni na x: -15 y: 0 olovka > clean up > 10 objavljujem događaj clean kad spazim događaj mati kad spazim događaj mati olovka učišča olovko 3 olovka postavi boja olovke olovka se u angreni 10 postavljaj 2 ponak 40 koraka olovka se 1 100 stopnjeneva ponak 22 koraka boja olovke 3 postavljaj 2 olovka se 1 40 stopnjeneva ponak 19 koraka olovka doga kreni na x: - slučajni broj od -210 do 220 slučajni broj od 30 do 60 kad spazim događaj mati olovka učišča olovko 3 olovka postavi boja olovke olovka se u angreni 90 postavljaj 20 ponak 11 koraka olovka se 5 2 stopnjeneva ponak 3 koraka boja olovke 3 postavljaj 2 olovka se 5 20 stopnjeneva ponak 10 koraka olovka doga kreni na x: - slučajni broj od -210 do 220 slučajni broj od 30 do -180
	Pozornica kad kliknem na zeleni trapez prebaci na kostim industry▼ kad spazim događaj grass▼ prebaci na kostim grass▼
	[Dodatni zadatak] Možete dopuniti ovu igru dodavanjem životinja koje će se pojaviti kada zrak više nije zagađen.
Alati i materijali za nastavnike	Snap! projekt "Drawing": https://snap.berkeley.edu/project?user=tadeja&project=Improve%20the%20climate
Alati i materijali za učenike	Programski jezik Snap!: https://snap.berkeley.edu/ Slike: grass.png, industry.png



Scenarij učenja 15 – Uhvati miša

Naziv scenarija	Uhvati miša
Potrebno predznanje iz programiranja	Dodavanje pozadine. Dodavanje objekata. Promjena kostima objektima. Dodavanje zvukova. Kretanje strelicama s tipkovnice uvažavajući ograničenja. Logički izrazi za dva različita stanja. Korištenje uvjeta.
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Beskonačna petlja● Slučajni broj● Brojilo● Mjerač vremena Ishodi učenja: <ul style="list-style-type: none">● Učenik će moći koristiti beskonačnu petlju za pomicanje objekta.● Učenik će moći koristiti slučajne brojeve za određivanje položaja objekta, pomicanje objekta za slučajan broj koraka i okretanje objekta za slučajan broj stupnjeva.● Učenik će moći koristiti brojač za brojanje miševa i koristi konačnu vrijednost kako bi rezimirao koliko je igrač bio uspješan.● Učenik će moći koristiti mjerač vremena za određivanje kraja igre.
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Programiranje igre u kojoj će igrač (mačka) morati uhvatiti miša. Zadatak: Programiranje aktivnosti u kojoj će mačka uhvatiti miša. Igrač mačku pomiče strelicama na tipkovnici, a miš se kreće nasumično. Kada mačka dodirne miša, miš će se sakriti i pojavit na nasumičnom mjestu. Također moramo imati i brojač koji će brojati koliko je puta mačka uhvatila miša. Za završetak igre također nam treba mjerač vremena. Na kraju igre djevojka mora rezimirati koliko je uspješan igrač bio, izgovorit će koliko je puta igrač uhvatio miša. Cilj: Učenik će se upoznati s konceptom dodjeljivanja više varijabli slučajnim vrijednostima. Naučit će kako koristiti blok <i>Operatori / slučajni broj od [x] do [y]</i> .
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje, zajedničko učenje, rješavanje problema, učenje zasnovano na igrama.
Oblici poučavanja	Frontalni rad Rad u paru / grupni rad



Razrada aktivnosti	<p>Motivacija – uvod</p> <p>Nastavnik pokazujući igru motiviram učenike. Razgovaram s učenicima kako bi započeli s programiranjem te igre. Zajedno s učenicima određujem redoslijed koraka, na primjer:</p> <ol style="list-style-type: none">1. Odrediti pozadinu i dodati objekt;2. Programirati mačku da se kreće tipkama sa strelicama;3. Programirati miša da se kreće nasumično;4. Programirati miša da se sakrije (i pojavi nasumično) kada dodirne mačku;5. Brojač programa;6. Dodati mjerač vremena i odredite kraj igre;7. Dodati djevojčicu i Programirati da rezimira koliko je uspješan igrač bio;8. Programirati djevojčicu da skoči kad dodirne miša;9. Dodati zvuk mačke / miša;10. ... <p>Učenici mogu pomoći u koracima ili sami smisliti svoja pravila igre (ali moraju slijediti podebljane korake).</p>  <p>Uvedimo operator za dodjeljivanje slučajnih varijabli</p> <p style="text-align: center;">slučajni broj od 1 do 10</p> <p>Učenici programiraju sljedeće zadatke u parovima/grupama uz pomoć učitelja.</p> <p>[Korak 1]</p> <p>Prvi korak je određivanje pozadine dane igrice. Učenici besplatno traže sliku na internetu. Zatim dodaju nove objekte - mačku i miša.</p>  
--------------------	---



[Korak 2]

Učenici programiraju mačku da se kreće pomoću strelica na tipkovnici. Tu moraju odrediti što se događa ako je mačka na rubu.



[Korak 3]

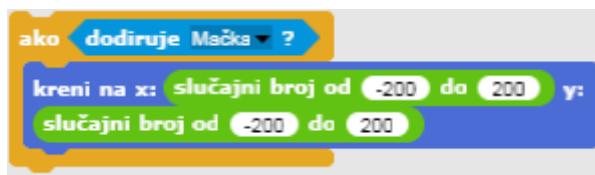
Učenik mora napraviti program u kojem se miš kreće nasumično. U našem slučaju, ideja je da miš beskonačnom petljom uzima slučajan broj koraka i zaokreće se za slučajan broj stupnjeva. Učenik to radi koristeći blok **kretanje / pomak [x] koraka** i blok **kretanje / okreni se [x] stupnjeva** u koji umeće operator **slučajan broj od [x] do [y]**.





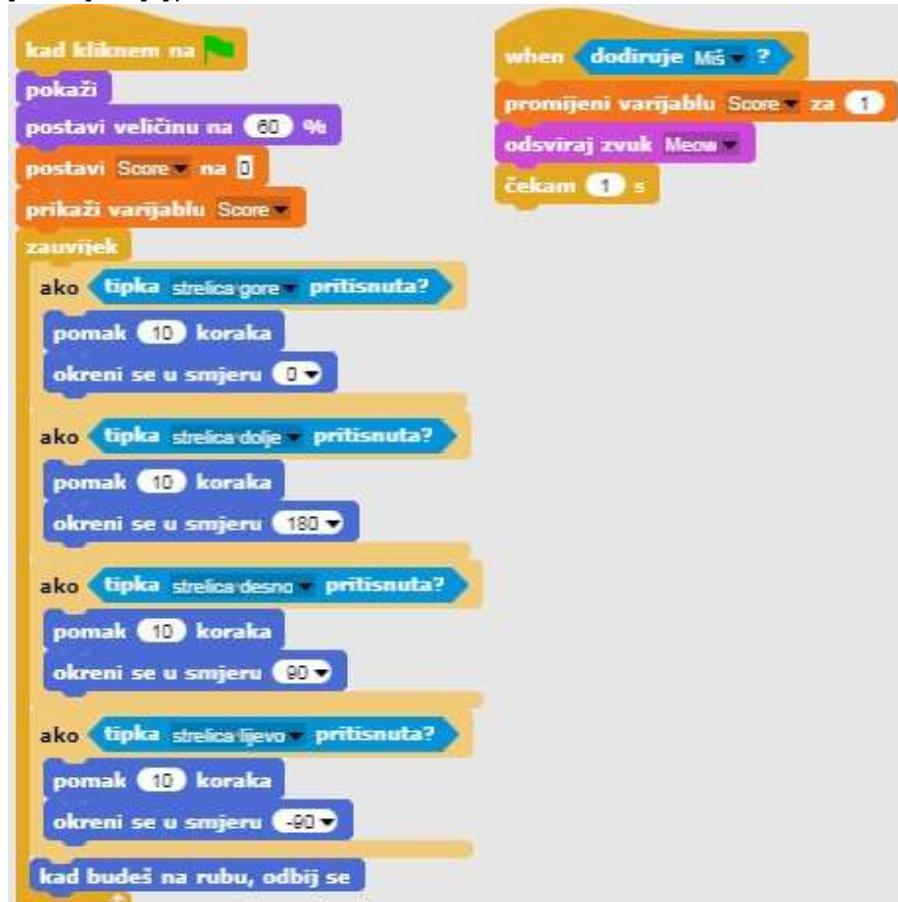
[Korak 4]

Sljedeći korak je programirati miša da se sakrije kada ga mačka dotakne. Ideja je da se miš sakrije i pojavi na slučajnoj lokaciji kada ga mačka dotakne. U ovom slučaju, igra ne završava nakon što prvi puta mačka uhvati miša. Učenici mogu dodati svoja vlastita pravila. U svakom slučaju moraju koristiti operator *slučajan broj od [x] do [y]*.



[Korak 5]

U slučaju da želimo znati broj uhvaćenih miševa, moramo dodati brojač. Učenik radi novu varijablu – score i dodaje ju u kod mačke. Score na početku igre uvijek mora biti jednak nuli. Učenici to rade pomoći bloku *variabla / postavi [varijabla] na [x]*. Ukoliko želimo da se rezultat prikazuje tijekom igranja, učenici moraju dodati blok *prikaz varijabla [varijabla]*. Zatim učenici dodaju novi kontrolni blok (*upravljanje / when*) kako bi provjerili dodiruje li mačka miša. Ako mačka dodiruje miša, rezultat se povećava za 1 (*variabla / promijeni [score] za [x]*).





[Korak 6]

Učenik definira kada igra završava. To rade dodajući mjerač vremena. Nakon nekog vremena (npr. 30 sekundi) miš i mačka nestanu, varijabla Score je skrivena i igra je završena.



Učenik dodaje taj kod u skriptu mačke i miša.

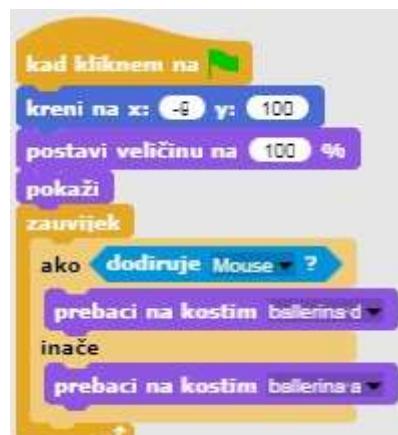
[Korak 7]

Učenik mora dodati naredbu kojom će djevojčica rezimirati koliko je igrač bio uspješan. Ako igrač ne uhvati niti jednog miša, djevojčica kaže: "Nisi uhvatio niti jednog miša!". Inače kaže: "Čestitam! Ulovila si x miševa!".



[Korak 8]

Učenik može dodati bilo koji element u svoju igricu. Na primjer, djevojčicu koja skače svaki put kada dotakne miša.





[Korak 9]

Učenici dodaju zvuk. Na primjer, mogu dodati zvuk mačke. Zvuk svira kada je miš ulovljen.

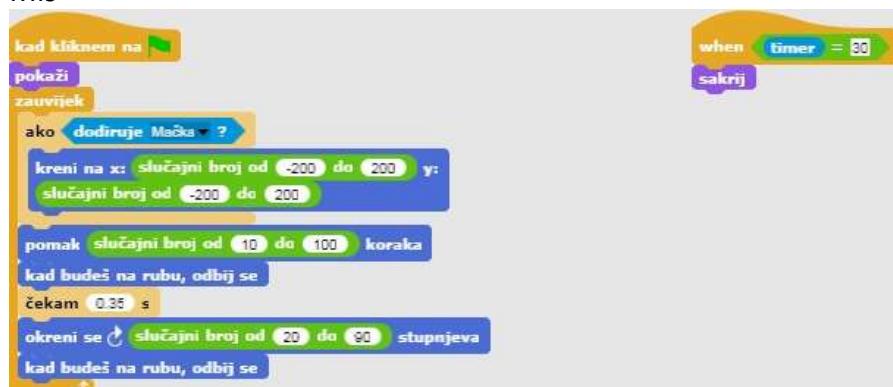


Učenici mogu doraditi kod i omogućiti sljedeće:

- Miš se pomiče od 20 do 60 koraka
- Kada mačka dotakne miša, miš se pomakne na lokaciju x=100
- Miš se zauvijek okreće za 90 stupnjeva
-

[Završni kod]

Miš



Djevojčica





	<p>Mačka</p> <p>Pozadina</p>
Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=tadeja&project=Catch%20the%20mouse Stranica s besplatnim slikama: https://pixabay.com/ Lajovic, S. (2011). Scratch. <i>Nauči se programirati in postani računalniški maček</i> . Ljubljana: Pasadena. Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i> . Ljubljana: MK.
Alati i materijali za učenike	Predložak izrađen u alatu Snap!: https://snap.berkeley.edu/project?user=tadeja&project=Catch%20the%20mouse_0 Stranica s besplatnim slikama: https://pixabay.com/



Scenarij učenja 16 – Kupnja hrane za piknik

Naziv scenarija	Kupnja hrane za piknik
Potrebno predznanje iz programiranja	Dodavanje teksta objektu Pokazivanje i skrivanje objekata Korištenje operatora Korištenje varijabli Korištenje stringova Korištenje uvjeta
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Variable● Uvjeti● Operatori Ishodi učenja: <ul style="list-style-type: none">● Učenik će koristiti varijable za postavljanje cijene za različite objekte● Učenik će mijenjati vrijednost varijabli jer se budžet mijenja kada igrač kupuje hranu● Učenik će koristiti naredbu <i>ako</i> za provjeru dostupnosti novca● Učenik će koristiti operatore za spajanje <i>tekst - vrijednost varijabli - tekst</i>● Učenik će koristiti operatore za usporedbu cijena i budžeta● Učenik će koristiti operatore (oduzimanje) za promjenu vrijednosti varijablama
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Djevojčica ide na izlet i treba joj pomoći oko kupnje hrane. Ima 15 kuna i ne može potrošiti više. Kada nešto kupi, vrijednost budžeta se promjeni. Ako joj je budžet prenizak ne može kupiti odabrano hranu. Zadatak: Učenici moraju programirati tri različita objekta: djevojčicu, hranu (koju mogu duplicitirati uz male promjene) i gumb za završetak. Djevojčica daje upute, govori koliko novaca igrač ima i na kraju (klikom na gumb za završetak) govori koliko je zdravih i nezdravih proizvoda igrač kupio. Prelaskom miša preko odabranog proizvoda ispisuje se njegova vrijednost te klikom na proizvod, ako igrač ima dovoljno novca, vrijednost budžeta se mijenja. Inače se hrana ne može kupiti. Cilj: Učenici će naučiti kako raditi sa varijablama: postavljanje različitih početnih vrijednosti, korištenjem uvjeta za usporedbu vrijednosti varijabli, promjenom vrijednosti varijabli, korištenjem varijabli za brojanje (ne) zdrave hrane. Osim toga, ponovit će dodavanje i spajanje teksta te naredbu <i>ako</i> .
Trajanje	45 minuta



Strategija i metode učenja i poučavanja	Aktivno učenje Game Based Learning – izrada igre Rješavanje problema
Oblici poučavanja	Individualni rad/rad u paru
Razrada aktivnosti	<p>Djevojčica je u trgovini i želi kupiti hranu za piknik. Ima 15 kuna. Prelaskom miša preko proizvoda može vidjeti cijenu hrane i kupiti je klikom. Hranu može kupiti samo dok ima dovoljno novaca. Klikom na gumb <i>Finish</i>, djevojčica govori koliko je zdravih i nezdravih proizvoda igrač kupio.</p>  <p>[Korak 1]</p> <p>Aktivnost je osmišljena za individualni rad ili rad u paru. Učitelj daje upute, objašnjava neke teže dijelove i pomaže kad je potrebno.</p> <p>Učenici biraju pozadinu i dodaju glavni objekt, npr. djevojčicu. Djevojčica daje upute na početku, npr.:</p>  <p>[Korak 2]</p> <p>Za ovu igricu trebat će nam nekoliko varijabli:</p> <ul style="list-style-type: none">• <i>budžet</i>, za postavljanje dostupnog novca,• <i>završi</i>, za završetak igre,• <i>zdrava_hrana</i>, za računanje koliko je zdrave hrane igrač kupio,



- *nezdrava_hrana*, za brojanje koliko je nezdrave hrane igrač kupio,
- varijabla za svaku hranu, npr. *lubenica_cijena*, za određivanje cijene pojedine hrane.

Na početku je varijabla budžeta postavljena na npr. 15 (Kn). Ostale tri varijable postavljene su na 0. Ovaj se kôd može dodati prije koda iz [Korak 1].



[Korak 3]

Učenici dodaju objekt (hranu) i biraju njegov kostim.
kôd hrane (lubenice) treba tri kontrolna događaja:

- a) *Kad se klikne zelena zastavica*: postaviti i prikazati cijenu hrane. Neka je cijena varijable razumno određena (naravno, ne 0, ali veća od 1).



- b) *Prelaskom miša preko proizvoda*: prikazati igraču koliko košta proizvod. Učenici mogu upotrijebiti blok *Izgleda-razmišljaj* koristeći spajanje *tekst - vrijednost varijable - tekst*, npr. :



- c) *Kad se klikne na određeni proizvod*
Postavljamo sljedeća pitanja:

- 1) U kojem slučaju igrač može kupiti proizvod, a u kojem ne?
- 2) Što se događa s budžetom kada igrač kupi hranu?
- 3) Kako računamo kupljene proizvode?
- 4) Što se događa s hranom na polici kada ju igrač kupi?

- 1) Igrač može kupiti proizvod ako ima dovoljno novaca. Na taj način

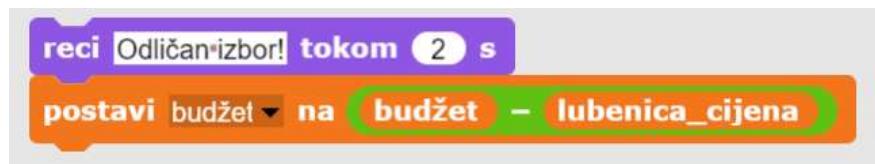


učenici moraju usporediti dvije varijable: *budžet* i *lubenica_cijena*. Ako lubenica košta više nego što je imao, ne može ju kupiti. Učenici mogu dodati neki tekst kako bi rekli igraču da ne može kupiti ovaj proizvod.



2) Ako igrač ima 15 Kn i kupi lubenicu za 4 Kn, sada ima $15 - 4 = 11$ Kn. Dakle, vrijednost budžeta je sada: *budžet - lubenica_cijena*.

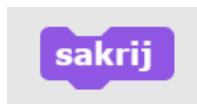
Učenici mogu i ovdje dodati neki tekst.



3) Prebrojavanje broja kupljenih proizvoda ostvarit će se promjenom varijable *zdrava_hrana* u 1.



4) Kad se hrana klikne, ona se sakrije.



Jedno moguće rješenje je:



[Korak 4]

Da bi na policama bilo više hrane, učenici mogu duplicitirati objekt lubenice. Recimo da će druga hrana biti torta. Kôd iz [Korak 3] tada treba neke izmjene. Učenici moraju:

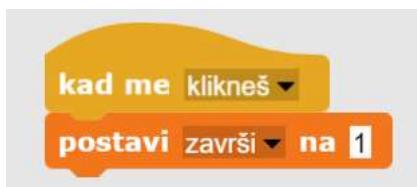
- promijeniti kostim
- napraviti novu varijablu: *torta_cijena*
- postaviti *torta_cijena* na neku vrijednost
- promijeniti u kodu svaki blok *lubenica_cijena* s *torta_cijena*
- promijeniti odgovor o kupnji torte
- zamijeniti *promjeni varijablu zdrava_hrana za 1* u *promjeni nezdrava_hrana za 1*.

Na primjer, kôd za kolač kada se klikne može biti:



[Korak 5]

Kada igrač završi kupnju, klikne na gumb *finish*. Da bismo rekli programu da je igrač kliknuo gumb (*finish*), varijablu *završi* postavimo na 1.



[Korak 6]



Na kraju se vraćamo u objekt da mu djevojčica kaže koliko je zdravih i nezdravih proizvoda kupio.

To će se učiniti provjerom je li igrač pritisnuo gumb *finish* – blokom *kada*. Ako je odgovor potvrđan, tada je vrijednost varijable *završi* 1, a djevojčica govori, npr. "Odabrali ste X zdrava proizvoda i Y nezdrava proizvoda".



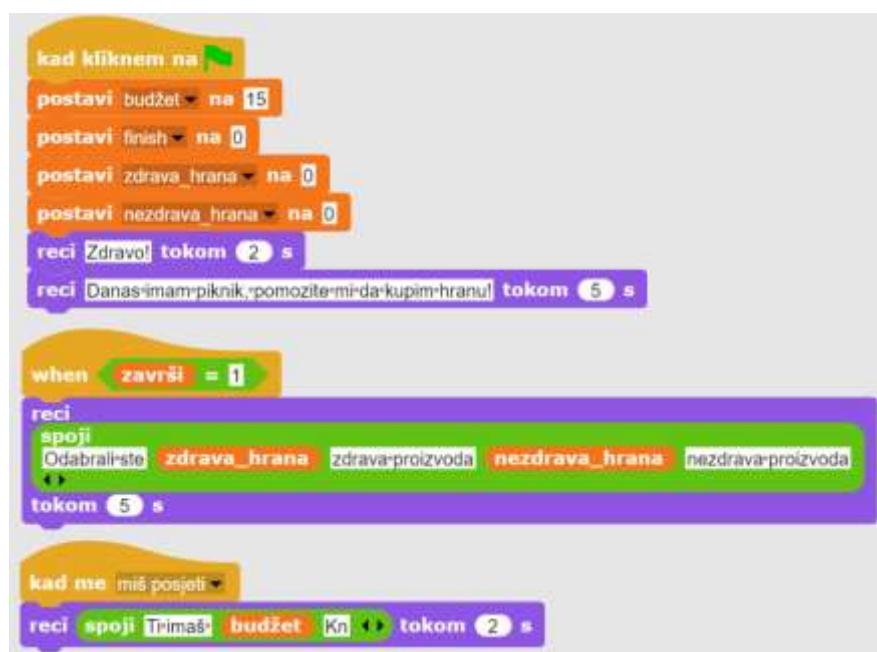
[Korak 7]

U bilo kojem trenutku tijekom igre, igrač može provjeriti njegov budžet mišem – postavljanjem miša na djevojčicu. Na primjer, ona može reći / razmišljati nešto poput:



[Cijeli kôd]

Djevojčica



Hrana



Gumb za završetak



[Dodatni zadatak]

Učenici mogu upotpuniti igru po želji ili prema sljedećim prijedlozima:

- Omogućiti kupnju svake namirnice 3 puta.
- Igraču na početku dodijeliti više novca.
- Ispisati poruku na kraju igre kojom će se prikazati što je igrač kupio.

Alati i materijali za nastavnike

Igra izrađena alatom Snap!:

<https://snap.berkeley.edu/project?user=mateja&project=Buying%20food%20for%20a%20picnic>

Primjer igre s dodatnim zadacima:

<https://snap.berkeley.edu/project?user=mateja&project=Buying%20food%20for%20a%20picnic%20%2B%20Add.%20Task>

Lajovic, S. (2011). *Scratch. Nauči se programirati in postani računalniški maček*. Ljubljana: Pasadena.

Vorderman, C. (2017). *Računalniško programiranje za otroke*. Ljubljana: MK.

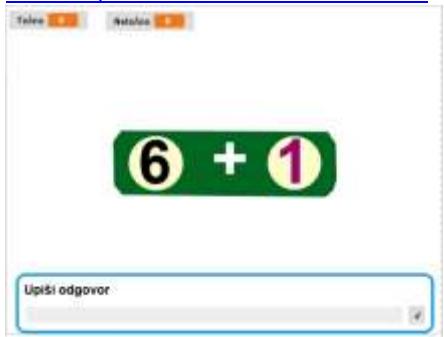
Alati i materijali za učenike



Scenarij učenja 17 – Operacije

Naziv scenarija	Operacije
Potrebno predznanje iz programiranja	Primjenjivanje varijabli za brojanje bodova i za promjenu kostima pozornice i objekta Korištenje slučajnog broja za postavljanje izgleda pozornice i kostima objekta Korištenje petlje <i>ponavljam</i> Korištenje uvjeta Primjena operacija uspoređivanja Korištenje <i>osjetila</i> za dijalog (pitaj...i čekaj) Korištenje bloka <i>objavljujem događaj</i> .
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Varijable● Uvjeti● Petlja● Blokovi <i>osjetila</i>● Objavljivanje događaja Ishodi učenja: <ul style="list-style-type: none">● Učenici će moći primjenjivati varijable za brojanje bodova, za promjenu izgleda pozornice i za kostime objekta● Učenici će moći inicijalizirati varijable za brojanje bodova● Učenici će moći koristiti uvjete i logičke operacije● Učenici će moći primjenjivati objavljivanje događaja za promjenu objekta i za računanje konačnog rezultata.
Cilj, zadaci i kratki opis aktivnosti	Opis aktivnosti: Za vrijeme igranja igrice, provjeravamo je li igrač usvojio aritmetičke operacije u Snap!-u. Pravila su sljedeća: 10 puta se bira aritmetička operacija slučajnim odabirom, prvi operand je broj 6, dok je drugi operand slučajni broj od 1 do 3. Igrač mora upisati točan odgovor. Broj točnih i netočnih odgovora se broji. Na kraju igrice izvještava se o konačnom rezultatu. Zadaci: Učenici trebaju izabrati scenarij, izgled pozornice i kostim objekta; osmislitи potrebite varijable, odreditи potrebite blokove. Na kraju je potrebno izraditi kod za pozornicu i za objekt. Dodatni zadatak bi mogao biti: Ovisno o rezultatu, dodijeliti objektu da kaže: „Bravo za tebe!“ ili „Nisi dobro usvojio aritmetičke operacije u Snap!-u“.



	<p>Cilj: Učenici će proširiti znanje o varijablama, slučajnim brojevima, petljama, objavljivanju događaja.</p>
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje (diskusija, eksperiment s unaprijed pripremljenom igrom) Učenje temeljeno na igram Učenje rješavanjem problema
Oblici poučavanja	Frontalni rad Individualni rad Rad u parovima
Razrada aktivnosti	<ol style="list-style-type: none">Nastavnik predstavlja problem u vezi potrebe igre da utvrди jesu li aritmetičke operacije u Snap!-u usvojene i demonstrira projekt. https://snap.berkeley.edu/snap/snap.html#present:Username=ddureva&ProjectName=operations3&editMode&noRun Nastavnik komentira kako oblikovati uvjet zadatka. Slučajnim odabirom se 10 puta izabire aritmetička operacija sa prvim operandom brojem 6 i drugim operandom slučajnim brojem od 1 do 3. Igrač mora upisati točan odgovor. Točni i netočni odgovori se broje. Na kraju igrice izvještava se o konačnom rezultatu.Objašnjavaju se varijable. Objasnjava se njihovo definiranje, inicijaliziranje i mijenjanje.Ponovo se prolazi kroz naredbe slučajnog broja, aritmetičke i logičke operacije, te kroz objavljivanje događaja.Raspravlja se o tome da li se glavni kod postavlja na skriptu od pozornice ili od objekta. U ovom primjeru, glavni kod je zapisan u skripti od pozornice, dok kod koji je zapisan u skripti od objekta ima za ulogu mijenjanje kosima i računanje konačnog rezultata.





	Scena	Izgled pozornice
	<pre> kad kliknem na [] postavi Netočno na 0 postavi Točno na 0 ponavljaj (10) postavi Operacija na slučajni broj od 1 do 4 prebaci na kostim Operacija objavljujem događaj Change costume number pitaj Upiši odgovor i čekaj ako ime kostima od Stage = - ako (6 - BrojKostima) = odgovor promijeni varijablu Točno za 1 inače promijeni varijablu Netočno za 1 ako ime kostima od Stage = + ako (6 + BrojKostima) = odgovor promijeni varijablu Točno za 1 inače promijeni varijablu Netočno za 1 ako ime kostima od Pozornica = ako (6 * BrojKostima) = odgovor promijeni varijablu Točno za 1 inače promijeni varijablu Netočno za 1 ako ime kostima od Pozornica = ako (6 / BrojKostima) = odgovor promijeni varijablu Točno za 1 inače promijeni varijablu Netočno za 1 objavljujem događaj Calculate result </pre>	

Kod za scenu sadrži inicijalizaciju varijabli za točan i netočan odgovor.

Za odabir operacije koriste se sljedeće naredbe:

```

postavi operation na slučajni broj od 1 do 4
prebaci na kostim operacija

```

Izbor kostima objekta se izvršava objavljuvanjem događaja objektu Broj. Izabrani broj kostima se pohranjuje u varijablu pod imenom BrojKostima koja je definirana za sve objekte projekta radi čega se ista varijabla može primjenjivati unutar koda za pozornicu.

Nakon što je izgled pozornice i kostim objekta izabrano slučajnim načinom, postavlja se pitanje igraču da upiše ispravan odgovor s obzirom

na operaciju naredbom: **pitaj Upiši odgovor i čekaj**.

Upisani odgovor se uspoređuje s ispravnim rezultatom ovisno o izabranoj operaciji.

Upotrijebljene su sljedeće naredbe:

Ako (uvjet)

Onda



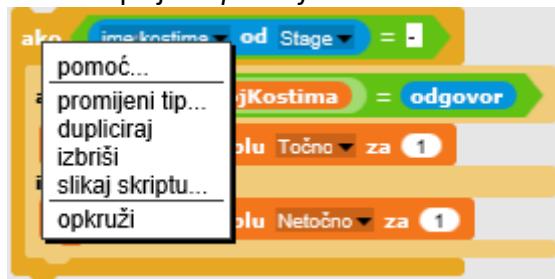
Ukoliko je operacija „-“ izabrana, provjerava se odgovora li razlika broja 6 i broja „BrojKostima“ igračevom upisanom rezultatu. Ukoliko se rezultati podudaraju, vrijednost varijable *Točno* se povećava za jedan, u suprotnom se povećava vrijednost varijable *Netočno*.



Isto vrijedi za ostale operacije.

S ciljem izbjegavanja ponovnog pisanja koda za preostale operacije, učenike se može naučiti kako kopirati kod (dio koda):

1. Desni klik na dio koda koji želiš kopirati
2. Izaberi opciju *dupliciraj*

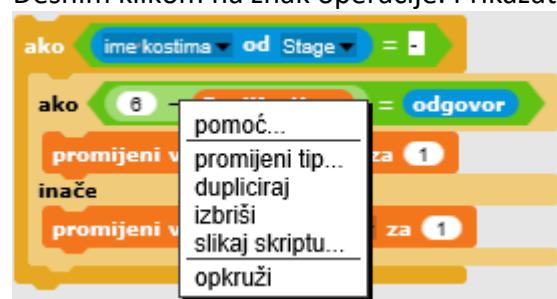


3. Koristi se mišem da duplikiranu kopiju koda smjestiš na odgovarajuće mjesto.

Nastavnik može biti diskretan i zadati učenicima da samostalno istraže kako kopirati dio koda.

Promjena operacijskog znaka.

1. Desnim klikom na znak operacije. Prikazati će se izbornik.





2. Izaberi *promijeni tip*. Prikazati će se lista operacija.



3. Izaberi operaciju.

Napomena: Ukoliko učenikove godine i znanje o aritmetičkim operacijama dopuštaju, onda se zadatak može proširiti tako da se uvede potenciranje (^) i dijeljenje s ostatom (mod).

Učenici rade u timovima tako da postave izgled pozornice i kostime objekta. Ukoliko se radi o zaostatku s vremenom, učenicima se može dati polugotov projekt u kojem su pozornica i izgled objekta unaprijed postavljeni.

Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=ddureva&project=operations3
Alati i materijali za učenike	Polugotova igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=ddureva&project=operations_half

**Scenarij učenja 18 – Recikliranje**

Naziv scenarija	Recikliranje
Potrebno predznanje iz programiranja	Prikazivanje i skrivanje objekta Korištenje varijabli za brojanje bodova Korištenje petlje zauvijek Korištenje uvjeta Korištenje operacija za usporedbu Korištenje osjetljivosti boja
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Varijable● Uvjeti● Petlje● Usmjeravanje u smjeru● Raspoznavanje blokova naredbi● Restrukturiranje koda Ishodi učenja: <ul style="list-style-type: none">● Učenici će moći koristiti naredbu <i>čekaj dok</i> i logičke operatore kako bi završili igru● Učenici će moći koristiti naredbu <i>čekaj dok</i> i blokiranje za promjenu pozornice● Učenici će moći koristiti varijable za brojanje bodova● Učenici će moći koristiti uvjete i logičke operacije● Učenici će moći usporediti kod sličnih objekata● Učenici će moći restrukturirati kod● Učenici će moći koristiti naredbe za pozicioniranje objekta
Cilj, zadaci i kratki opis aktivnosti	Zadatak: Učenici moraju istražiti kod pozadine i objekata, usporediti kod objekata otpadnog papira i stakla, dodati nove objekte i skripte, promjeniti postojeću skriptu pozornice s obzirom na novo dodane objekte. Dodatni zadatci: <ul style="list-style-type: none">● Promjeniti položaj objekata otpada slučajnim odabirom koordinata objekta● Smanjiti broj razina (pozornica) i izdvojiti robota kao zaseban objekt (robot je dio pozadine pozornice) Cilj: Učenici će poboljšati prethodno stečeno znanje te će proširiti scenarij igre sa novim objektima, kodom i promjenom koda s obzirom na nove objekte. Učenici će moći restrukturirati kod.
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Game Based Learning – igra Razgovor Demonstracija Rješavanje problema



Oblici poučavanja	Frontalni rad Rad u paru Individualni rad
Razrada aktivnosti	<p>Učitelj postavlja problem razdvajanja otpada i komentira boje kanti za različite vrste smeća – plava za papir, zelena za plastiku.</p> <p>Učitelj zadaje učenicima da odigraju igru i opišu je riječima: Koliko pozornica gledaju i koliko objekata postoji (likova)? Kako igra počinje? Koji objekt traži ime igrača? Koliko varijabli se koristi i kako se zovu? Što se događa kada se papir stavi u spremnik za staklo, a što kada se stavi u spremnik za papir?</p> <p>The image shows a Scratch script. At the top, there are three buttons: 'Staklo' (Glass) with value 0, 'Papir' (Paper) with value 0, and 'Ime' (Name). A grey robot with red arms and a blue head asks 'Pozdrav, kako se zoveš?' (Hello, what's your name?). Below the robot are two recycling bins: a green one on the left and a blue one on the right, both labeled 'We Recycle'. At the bottom is a text input field with a checkmark icon.</p> <p>[Korak 1] Ažuriranje proučenih naredbi</p> <p>Poziva se naredba za dijalog s korisnikom (igračem). Dodaje se komentar o promjeni scena – Scena 1 s robotom, Scena 2 sa školom i smećem te Scena 3 s robotom i natpisom 'Bravo!'. Raspravlja se o mogućim naredbama za promjenu scene.</p>



Raspravlja se da provjeru ispravnog odlaganja otpada u spremnike treba provesti pomoću uvjetnog bloka i blokova s uvjetima iz skupine Sensing (dodir).

Daje se usmeni opis: ako komad papirnatog otpada dodirne spremnik za papir, otpad se sakrije (odložen je u odgovarajući spremnik) te se dodaje 1 bod za skupljeni papirnatni otpad. Ako papirnati otpad dotakne spremnik za staklo, pojavljuje se poruka: "Ovo nije spremnik za papir". Isto se događa sa staklenim otpadom.



[Korak 2]

Proučavanje koda scena i objekata

Nakon rasprave o mogućim rješenjima problema, raspravlja se o kodu za scenu i objekte.

Kod scene komentira se s naglaskom na:

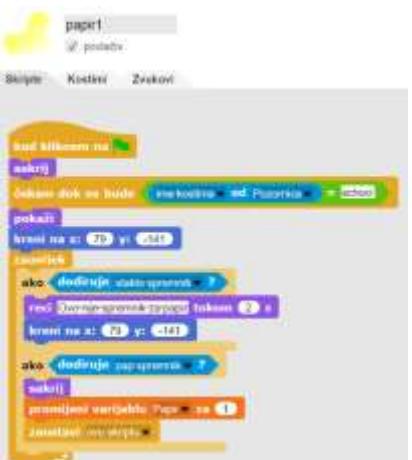
- Postavljanje početne vrijednosti varijable imena i korištenje iste u dijalogu sa korisnikom;
- Promjena scenskog prizora (pozornice) i uvjet za završetak igre.





Kada gledate kod objekata, preporučljivo je prikazati ih na jednoj stranici ili po dva ispisana koda za papirnati i stakleni otpad. Radi se usporedba između sličnih i različitih elemenata u kodu.











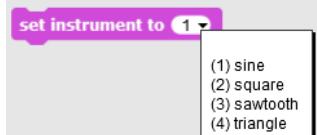
	<p>[Korak 3]</p> <p>Postavljanje zadatka za dovršetak igre s dva nova objekta – papirnati otpad i stakleni otpad, dodijelimo im kod te promijenimo scenu (pozornicu) i kod spremnika za otpad. Raspravlja se o dva nova objekta.</p> <p>Opcije – Umnožite postojeće objekte i uredite ih u Snap!-u, kreirajte nove u grafičkom uređivaču ili pretražite besplatne slike na internetu i uvezite ih u igru.</p> <p>Potrebno je komentirati i promjene koda scene s obzirom na završetak igre. Treba se prokomentirati i postavljanje početnih vrijednosti varijabli u kodu scene, a ne u kodu spremnika, te napraviti potrebnu prilagodbu.</p> <p>Po potrebi, učitelj može otežati zadatku:</p> <ul style="list-style-type: none">• Otpad treba rasporediti na bilo koje prikladno mjesto prilikom pokretanja igre. Treba napomenuti da koordinate unutar kojih se otpad može rasporediti moraju biti ograničene, tako da otpad bude raspoređen na realnom mjestu. Na primjer, koordinate su ograničene crvenim pravokutnikom.  <ul style="list-style-type: none">• Uvođenje novog objekta za robota i smanjenje broja elemenata na pozornici. Napisati odgovarajući kod za robota kako bi mogao sudjelovati u razgovoru sa korisnikom (igračem) umjesto objekta plavog spremnika. <p>Alati i materijali za nastavnike</p> <p>Igra izrađena u alatu Snap!: https://snap.berkeley.edu/project?user=ddureva&project=recycling</p> <p>Alati i materijali za učenike</p> <p>Igra izrađena u alatu Snap!: https://snap.berkeley.edu/project?user=ddureva&project=recycling</p>



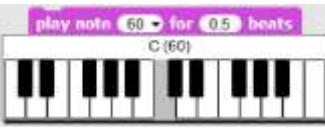
Scenarij učenja 19.1 – Sviranje klavira

Naziv scenarija	Sviranje klavira
Potrebno predznanje iz programiranja	Korištenje varijabli za brojanje bodova Korištenje događaja <i>Kada sam pritisnut</i> Korištenje petlje za ponavljanje Korištenje uvjeta Korištenje emitiranih događaja za promjenu scenarija/uređivanje pozornice i za upravljanje aktivnostima objekata
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">• Varijable• Uvjeti• Petlje• Emitiranje događaja• Zvukovi• Programiranje glazbe Ishodi učenja: <ul style="list-style-type: none">• Učenici će moći koristiti varijable za brojanje bodova• Učenici će moći inicijalizirati varijable za brojanje bodova• Učenici će moći koristiti uvjet za procjenu postignutih bodova• Učenici će moći koristiti događaj emitiranja za promjenu scenarija / uređivanje pozornice / i za aktivnosti objekata• Učenici će moći koristiti blokove iz grupe <i>Zvuk</i> da bi komponirali melodije• Učenici će moći prepoznati potrebu za korištenje petlje za ponavljanje da bi se smanjio broj blokova u skriptama• Učenici će moći proširiti funkcionalnost igre
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Uđimo u prekrasan svijet kraljice Marije. Ona poziva igrača u svoju palaču da posluša neku glazbu. U plesnoj dvorani njezin mali dinosaurus Dino svira klavir. U igri Dino svira nekoliko glazbenih tonova i igrači moraju prepoznati o kojem tonu se radi. Ako pogode, dobivaju jedan bod za ispravan odgovor, a ako ne znaju, oduzme im se jedan bod za pogrešan odgovor. Nakon prepoznavanja tonova postavlja se složeniji zadatak: Dino svira melodiju, a igrač mora prepoznati o kojoj se pjesmi radi. Igrač dobiva 5 bodova za pravilno prepoznatu pjesmu. Zadatak: Učenici koriste djelomično gotovu datoteku s scenografijom / scenskom dekoracijom / i kostima objekata. Moraju planirati potrebne varijable, odrediti koji su blokovi potrebni; upoznati se s blokovima grupe <i>Zvuk</i> i načinom sviranja nota. Stvaraju skripte za reprodukciju nekoliko melodija. Cilj: Studenti će upoznati kodiranje i sviranje melodija te će



	poboljšati svoje prethodno stečeno znanje o varijablama, petlji, uvjetnim, emitiranim i ostalim događajima.
Trajanje	90 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje (diskusije, eksperimentiranje s prethodno pripremljenom igrom) Game Based Learning - igra Rješavanje problema
Oblici poučavanja	Individualni rad / Rad u paru / Frontalni rad s cijelim razredom
Razrada aktivnosti	<p>Učitelj postavlja zadatak stvaranja igre. Raspravlja se o sredstvima pomoću kojih se zadatak može dovršiti. Zaključeno je da trenutno nisu svjesni dostupnih resursa za pisanje koda za sastavljanje melodije.</p> <p>Učitelj demonstrira dio igre skladajući napjev.</p> <p>https://snap.berkeley.edu/project?user=ddureva&project=Play_a_Piano_1</p>  <p>Učitelj pokazuje kôd i objašnjava kako se mogu koristiti naredbe grupe Zvuk. U programu Snap! mogu se koristiti zvukovi iz ugrađene biblioteke, kao i datoteke s računalna ili glazbeni tonovi koji se sviraju na različitim instrumentima.</p> <p>Za odabir alata koristite naredbu:</p>  <p>Učenici testiraju zvuk raznih instrumenata.</p> <p>Učitelj objašnjava način postavljanja glazbenih tonova:</p> <p>Koristi se naredba . U njemu prvi broj postavlja ton, a drugi broj opisuje koliko dugo će se ton reproducirati.</p> <p>Kada kliknete strelicu pokraj prvog broja, pojavljuje se tipkovnica za</p>



	<p>klavir i iz nje se može odabrati ton. Ova klavirska tipkovnica obuhvaća dvije oktave.</p>   <p>Trajanje svake bilješke postavlja se brojevima 1 - cijela nota, 0,5 - polovinka, 0,25 - četvrtinka. (Za učenike ne znaju, decimalni se broj može prikazati u obliku razlomaka: $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ itd.)</p>  <p>Prema navođenju učitelja, učenici mogu eksperimentirati sa naredbama.</p> <p>Raspravlja se o scenariju melodije Jingle Bells (Zvončići).</p>  <p>Zadatak je postavljen tako da se smanji broj redaka u kodu koji se ponavljaju. Raspravlja se o naredbi koja će se koristiti (petlja za ponavljanje). Učenici su podijeljeni u timove koji su potrebni za kreiranje igre postavljene na početku predavanja. Svaki tim raspravlja o scenariju igre i opisuje plan igre (detaljan opis objekata i aktivnosti na pozornici). Može se dodati uvjet da dinosaurus može plesati tijekom sviranja. (Dinosaurus ima nekoliko kostima za koje su unaprijed pripremljene datoteke).</p>
Alati i materijali za nastavnike	<p>Igra izrađena alatom Snap!:</p> <p>https://snap.berkeley.edu/project?user=ddureva&project=Play_a_Piano_1</p> <p>https://snap.berkeley.edu/project?user=ddureva&project=PlayAPiano_no</p>
Alati i materijali za učenike	<p>Polugotova igra izrađena alatom Snap!:</p> <p>https://snap.berkeley.edu/project?user=ddureva&project=Play_a_Piano_Half_backed</p>



Scenarij učenja 19.2 – Sviranje klavira

Naziv scenarija	Sviranje klavira
Potrebno predznanje iz programiranja	Korištenje petlji za ponavljanje Korištenje varijabli Korištenje uvjeta
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">● Uvjeti● Petlje Ishodi učenja: <ul style="list-style-type: none">● Učenik će moći koristiti petlje za ponavljanje za reprodukciju glazbe● Učenik će moći koristiti kôd kako bi objekti reagirali na unos● Učenik će moći dodati zvukove objektu● Učenik će moći koristiti kôd za promjenu kostima objektu
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Učenik mora odsvirati pjesmu na klaviru u skladu s danim notama. Zadaci: Učenici trebaju programirati tipke za klavir - svaka tipka mora svirati određeni ton. Na pozornici se moraju pokazati dva različita gumba, jedan za prikaz nota, a drugi za reprodukciju melodije. Cilj: Učenici će naučiti svirati melodiju i mijenjati kostim klikom na objekt.
Trajanje	45 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje Game Based Learning - Igra Rješavanje problema
Oblici poučavanja	Individualni rad / Rad u parovima
Razrada aktivnosti	Na početku se na pozornici prikazuje klavir. Pored klavira trebaju biti dva gumba. Klikom na prvi gumb trebaju se prikazivati note i riječi pjesme, a klikom na drugi gumb treba svirati melodija koju je potrebno ponoviti. Uz to, pored glasovira bi trebao biti gumb "X", koji će ponovno pokrenuti projekt. [Korak 1] Preuzmite datoteku s materijalima. Datoteka sadrži pozadinu i sve objekte potrebne za ovaj zadatak. Na pozornicu umetnite praznu pozadinu iz materijala. 



Dodajte novi objekt-tipku C i uvezite kostim "C" iz materijala.



Učinite isto za tipke D, E, F, G, A, B i stavite ih jednu za drugom.



Dodajte novi objekt – crnu tipku 1 i uvezite kostim "black_key" iz materijala. Kopirajte ove objekte 4 puta da biste dobili 5 crnih tipki i imenujte ih black key 2 do black key 5. Crne tipke moraju biti između svake dvije bijele tipke, osim E, F.



Poništite gumb "draggable", tako da se tipke ne mogu pomicati tijekom reprodukcije.

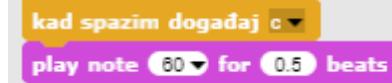


[Korak 2]

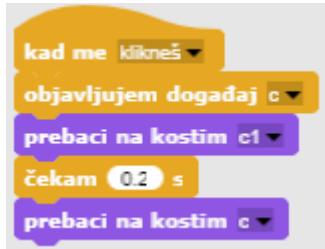
Omogućite reprodukciju tonova pritiskom na tipke. Za tipku "C" dodajte šešir "Kad pritisnem tipku" i dopustite mu da emitira poruku "c".



Za stvaranje zvuka kad pritisnete tipku, dodajte šešir na blok "Kad spazim događaj c" i odsviraj ton 60 sa 0.5 otkucaja.



Kako biste istaknuli koja se tipka pritisne, kostim tog objekta treba privremeno promijeniti. Uvezite c1 kostim u objekt C. U bloku "Kad me klikneš" promijenite kostim u c1 za 0,2 sekunde, a zatim se vratite u kostim c.



[Korak 3]

Ponovite korak 2 za ostale bijele tipke. Definirajte da ostale tipke sviraju sljedeće tonove:

D - 62

E - 64

Ž - 65

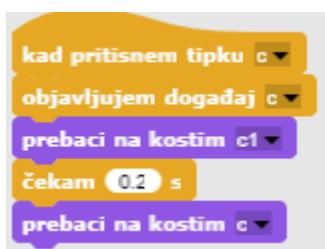
G - 67

A - 69

B - 71

[Korak 4]

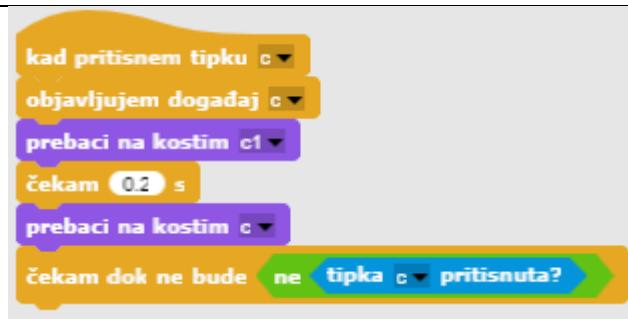
Za reprodukciju klavira pomoću tipkovnice, dodajte blok "Kad pritisnem tipku c" na tipku c objekta i kopirajte ostatak koda iz bloka "Kad me klikneš".



Opazite da ako je pritisnuta tipka c na tipkovnici, zvuk će se ponavljati sve dok držite pritisnutu tipku. To se događa jer se poruka "c" ponovno emitira. Kako biste prestali emitirati poruku, na kraju koda dodajte blok "čekam dok ne bude" s kontrolne ploče.



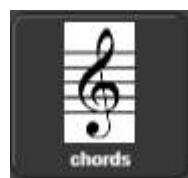
Za završetak emitiranja poruke koristite operatora "ne" i dodajte im blok "tipka c pritisnuta".



Učinite isto i za preostale bijele tipke.

[Korak 5]

Napravite novi objekt i uvezite sliku violinskog ključa kao kostim. Ovo će biti gumb za prikaz riječi i bilješki koje se trebaju svirati.

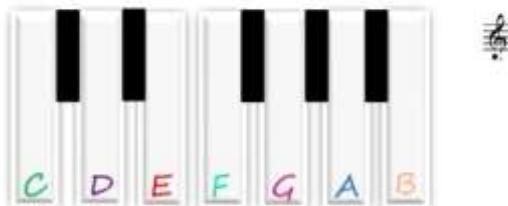


Da biste prikazali bilješke, omogućite emitiranje poruke "chords" kad se klikne gumb.



Uvezite novi kostim „chords“ za pozornicu.

TWINKLE TWINKLE LITTLE STAR HOW I WONDER WHAT YOU ARE
C C G G A A G F F E E E D D P P C
UP ABOVE THE WORLD SO HIGH LIKE A DIAMOND IN THE SKY
G G F F E E D D G G F F E E E D
TWINKLE TWINKLE LITTLE STAR HOW I WONDER WHAT YOU ARE
C C G G A A G F F E E E D D P P C



Dodajte kôd koji omogućava pozornici da promijeni kostim u "chords" kad primi poruku "chords".



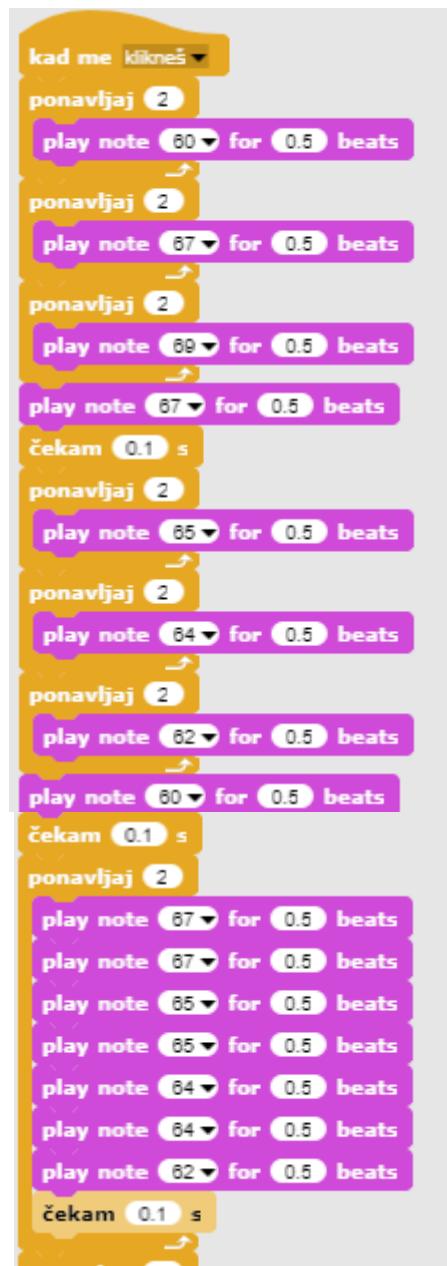


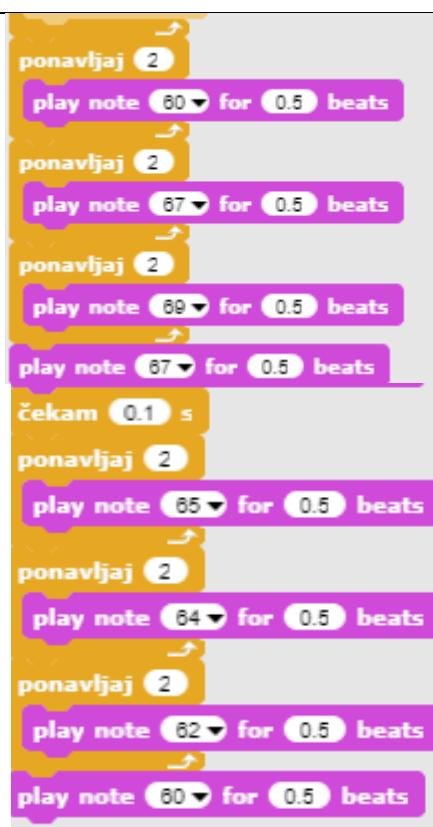
[Korak 6]

Napravite novi objekt i uvezite sliku tipke s notama kao kostim. Ovo će biti gumb za reprodukciju pjesme koju je potrebno ponoviti.



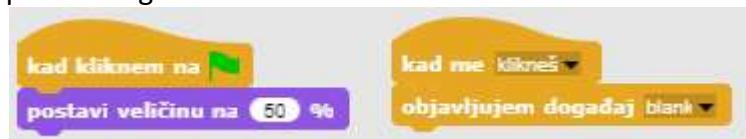
Dodajte blok "Kad me klikneš", a nakon toga bilješke koje će se reproducirati. To je ista pjesma koja je prikazana u notama.



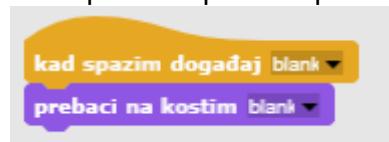


[Korak 7]

Stvorite novi gumb X koji će resetirati projekt (bez napomena). Stvorite novi objekt - resetirajte, odaberite kostim "X" i postavite njegovu veličinu na 50%. Omogućite emitiranje "prazne" poruke kada pritisnete gumb.



Dodajte blok "Kad spazim događaj" na pozornicu da promijeni kostim u "prazno" nakon primitka poruke "prazno".





[Konačni kod]

C tipka

```
when green flag clicked
  [when key c pressed v
    say [C tipka v
    wait (0.2) seconds
    change costume to [costume1 v
    when key c pressed v
      say [C tipka v
      wait (0.2) seconds
      change costume to [costume2 v
      when spazim dogadaj c v
        play note [60 v for [0.5] beats
      end
    end
  end
end
```

Violinski ključ

```
when green flag clicked
  [when me clicked v
    say [kad me klikneš v
    objavljujem dogadaj chords v
  end
end
```

Nota

```
when green flag clicked
  [when me clicked v
    ponavljaj (2) [
      play note [60 v for [0.5] beats
    ]
    ponavljaj (2) [
      play note [67 v for [0.5] beats
    ]
    ponavljaj (2) [
      play note [69 v for [0.5] beats
    ]
    play note [67 v for [0.5] beats
    čekam (0.1) seconds
    ponavljaj (2) [
      play note [65 v for [0.5] beats
    ]
    ponavljaj (2) [
      play note [64 v for [0.5] beats
    ]
    ponavljaj (2) [
      play note [62 v for [0.5] beats
    ]
    play note [60 v for [0.5] beats
  ]
end
```



```
čekam 0.1 s
ponavljam 2
  play note 67 for 0.5 beats
  play note 67 for 0.5 beats
  play note 65 for 0.5 beats
  play note 65 for 0.5 beats
  play note 64 for 0.5 beats
  play note 64 for 0.5 beats
  play note 62 for 0.5 beats
čekam 0.1 s
ponavljam 2
  play note 60 for 0.5 beats
  ponavljam 2
    play note 67 for 0.5 beats
    ponavljam 2
      play note 69 for 0.5 beats
      play note 67 for 0.5 beats
    ponavljam 2
      play note 65 for 0.5 beats
    ponavljam 2
      play note 64 for 0.5 beats
      play note 62 for 0.5 beats
      play note 60 for 0.5 beats
```

X

```
kad kliknem na flag
postavi veličinu na 50 %
kad me klikneš
objavljujem događaj blank
```

Pozornica

```
kad spazim događaj chords
prebac na kostim chords
kad spazim događaj blank
prebac na kostim blank
```



	<p>[Dodatni zadatak]</p> <p>Učenici mogu upotpuniti igru po želji ili koristeći sljedeće prijedloge:</p> <ul style="list-style-type: none">• Duplicirajte objekt Nota, promijenite joj poziciju i napišite program za drugu pjesmu.• Dodajte pozadinu s notama za drugu pjesmu.
Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=ifrankovic&project=Play%20a%20Piano
Alati i materijali za učenike	Polugotova igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=ifrankovic&project=Play%20Piano Slike: Slike objekata: <ul style="list-style-type: none">• a.png, a1.png• b.png, b1.png• c.png, c1.png• d.png, d1.png• e.png, e1.png• f.png, f1.png• g.png, g1.png• black_key.png, violin_key.png, note.png Pozadine: blank.png, notes.png

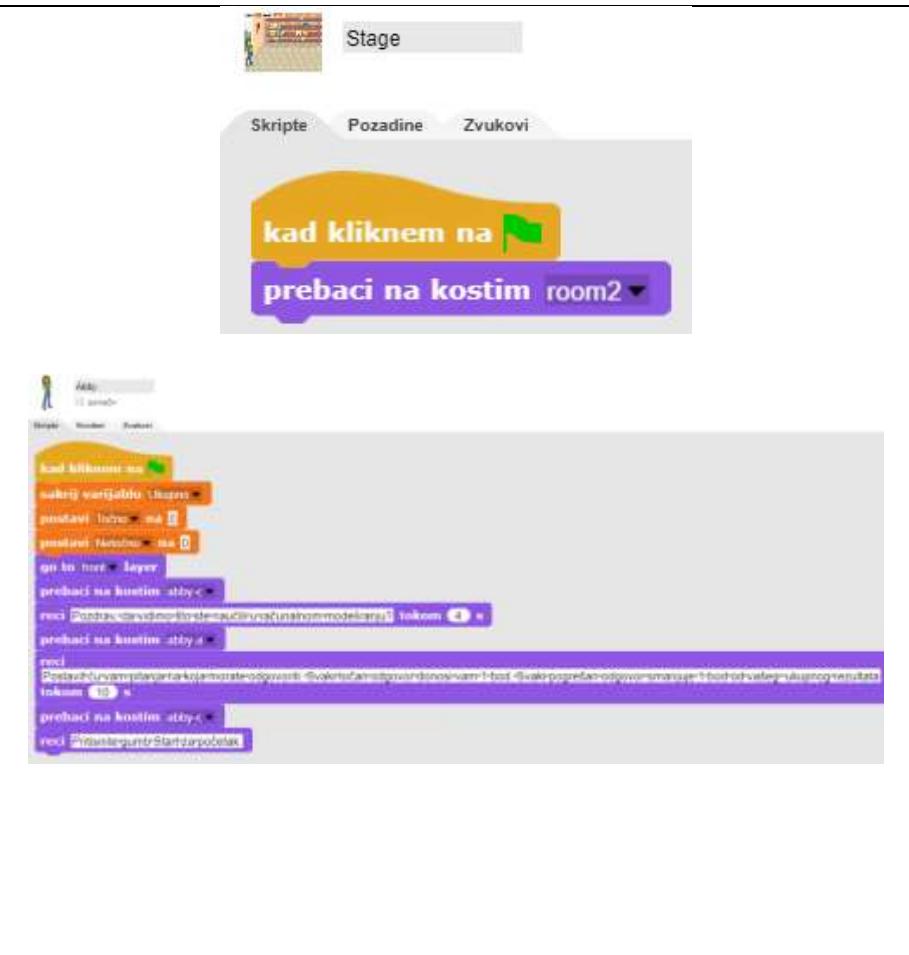
**Scenarij učenja 20 – Test**

Naziv scenarija	Test
Potrebno predznanje iz programiranja	Prikaz i skrivanje objekata Korištenje varijabli za brojanje bodova Korištenje petlje zauvijek Korištenje uvjeta Korištenje operacija za usporedbu Upotreba blokova Očitanja Promjena faze
Ishodi učenja	Koncepti programiranja: <ul style="list-style-type: none">• Variable• Uvjeti• Petlja• Blokovi Očitanja Ishodi učenja: <ul style="list-style-type: none">• Učenik će koristiti uvjet za procjenu odgovora - točno ili netočno• Učenik će koristiti blokove za promjenu pozornice u kostimu• Učenik će koristiti varijable za brojanje bodova• Učenik će koristiti logičke operacije• Učenik će koristiti vanjski grafički uređivač za pripremu složenih pozadina pozornica.
Cilj, zadaci i kratki opis aktivnosti	Kratki opis: Pomozite svom učitelju da testira vaše znanje o Snap!-u stvaranjem testa za provjeru znanja o naredbama. Zadatak: Učenici moraju istražiti primjer igre, izabrati iz „polugotovih“ igara, pronaći ili osmislići vlastiti objekt (<i>sprite</i>) koji će postavljati pitanja ili osmislići pozadinu početne pozornice i pozadinu pozornice s odgovarajućim pitanjima, izmijeniti i proširiti skripte u testu s obzirom na pitanja. Cilj: Učenici će poboljšati svoja stečena znanja i proširiti će scenarij igre s novom pozadinom, kodom i promjenom koda s obzirom na nove pozornice.
Trajanje	90 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje (rasprava, eksperimentiranje s unaprijed pripremljenom igrom) Game Based Learning – izrada igre Rješavanje problema
Oblici poučavanja	Frontalno poučavanje Individualni rad/ rad u paru



Razrada aktivnosti	<ol style="list-style-type: none">1. Učitelj postavlja problem potreban za stvaranje igre-testa za provjeru znanja programiranja.2. Zadaje učenicima da igraju igru i opisuju riječima: Koliko scena mogu uočiti, a koliko objekata (likova)? Kako igra počinje? Koliko se koristi varijabli, kako se imenuju, za što se koriste? Što se događa kada je odgovor točan/netočan? Kako su pitanja predstavljena u testu? Po izboru učitelja može se koristiti individualni rad ili rad u parovima.3. Komentirajte algoritam postavljanja i odgovora na pitanja.<ul style="list-style-type: none">• Prelazak na scene (sadrži pitanje)• Dodjela Abby kostima za postavljanje pitanja• Abby kaže - Odgovorite Da ili Ne• Igrač unosi odgovor - Da ili Ne• Ako je odgovor točan, Abby kaže "Točno", a broj točnih odgovora se povećava; Inače Abby kaže "Netočno" i broj pogrešnih odgovora se povećava.4. Komentirajte što se događa nakon što odgovorite na sva pitanja.<ul style="list-style-type: none">• Promjena kostima / pozadine na pozornici;• Abby pokazuje broj ispravnih i pogrešnih odgovora i daje ukupnu procjenu.5. Ispitivanje kodova u igri/Modificiranje postojećeg znanja Komentiraju se uvjetne naredbe, naredbe za vođenje dijaloga s korisnikom, za promjenu scena te kostima lika. Pregledavaju se kodovi svakog lika, a stvaranje varijable se komentira.
---------------------------	---



	 <p>The image shows the Scratch interface. At the top, there's a stage area with a small character and a "Stage" button. Below the stage are tabs for "Skripte", "Pozadine", and "Zvukovi". The main area is the script editor, which contains the following script:</p> <pre>when green flag clicked [set costume to room2 v] [say [Prehod na kostim abyc] for [all turtles] for [4 sec]] [say [Prehod na kostim abyc] for [all turtles] for [4 sec]]</pre> <p>The script consists of two "repeat" loops. Each loop contains a "set costume to room2 v" command and a "say [Prehod na kostim abyc] for [all turtles]" command repeated four times.</p>
--	--



Komentiraju se situacije kada je točan odgovor *Da* i kada je točan odgovor *Ne*.

Kôd za ocjenjivanje detaljno je obrađen, kao i zašto se koristi varijabla *Ukupno*.



Raspravlja se o načinu oblikovanja scenske pozornice za pojedinačna pitanja.

U Snap! nije moguće pisati tekst u kostimima i u sceni pa je potrebno koristiti vanjski grafički uređivač. Druga mogućnost je korištenje MS PowerPoint-a za stvaranje pitanja i izvoz odgovarajućeg tekstualnog okvira u grafičkom obliku.

Umetanje kostima u Snap! se može ponoviti.

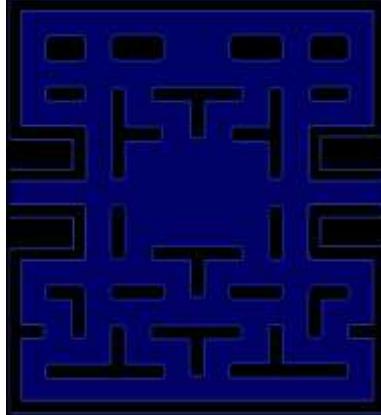


	<p>Proces izrade testa:</p> <ol style="list-style-type: none">1. Podjela grupe na timove od 2 ili 3 učenika.2. Objavljanje teme za testna pitanja. Na primjer - korištenje varijabli, petlje, kretanje, očitanja, aritmetičke i logičke operacije.3. Dizajniranje scena s pitanjima o temi od strane odgovarajućeg tima. Ako je potrebno, nastavnik savjetuje učenike o sadržaju pitanja. O pitanjima se raspravlja i svaki tim stvara scenu s najmanje dva pitanja.4. Izrada koda. Učenicima je na raspolaganju biblioteka s kostimima pozornice i objekata. Ako žele, mogu stvoriti i vlastite slike/kostime..
Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=ddureva&project=test2
Alati i materijali za učenike	Polugotova igra izrađena alatom Snap!: https://snap.berkeley.edu/snap/snap.html#present:Username=spelac&ProjectName=C4G_20_test_en_tmp

**Scenarij učenja 21 – Pojednostavljena igra PACMAN**

Naziv scenarija	Pojednostavljena PACMAN igra
Potrebno predznanje iz programiranja	Dodavanje pozadine Dodavanje novog objekta Definirati što objekt govori Kretanje objekata Slučajni broj
Ishodi učenja	<p>Koncepti programiranja:</p> <ul style="list-style-type: none">● Kretanje objekta obzirom na događaj● Osjetilo jedne boje● Čitanje logičkih vrijednosti u logičkim izrazima● Definiranje, razlikovanje, dinamičko ispitivanje i odgovaranje na dva različita stanja u igri <p>Ishodi učenja:</p> <ul style="list-style-type: none">● Učenik će moći omogućiti kretanje objekta tipkama sa strelicama koristeći događaje i uzimajući u obzir ograničenja● Učenik će moći koristiti blok osjetilo boje kako bi dobio logičku vrijednost za očitavanje osjetila jedne boje.● Učenik će moći prepoznati da stanje objekta može biti izraženo bojom koju objekt dodiruje● Učenik će moći razlikovati dva različita stanja i izražava ih logičkim izrazima● Učenik će moći koristiti petlju za ponovnu promjenu trenutnog stanja kada se položaj objekta dinamički mijenja● Učenik će moći upotrebljavati ako – inače grananje za postavljanje različitih odgovora na osnovu trenutnog položaja objekta
Cilj, zadaci i kratki opis aktivnosti	<p>Kratki opis: Programska igra u kojoj će glavni lik sakupljati nasumično postavljene zvijezde i pritom biti proganjan duhom.</p> <p>Zadaci: Učenici moraju programirati kretanje glavnog lika kako bi se on kretao unutar labirinta. Moraju provesti ograničenja kretanja kako se glavni lik ne bi mogao kretati kroz zidove. Zatim, moraju programirati objekt zvijezdu koja će se klonirati kada igra započne, a zatim pojaviti nasumično na novom mjestu svaki put kada ju glavni lik prikupi. Moraju pohraniti vrijednosti prikupljenih zvijezda te završiti igru kada igrač sakupi 20 zvijezda. Kako bi igra bila zanimljivija, učenici moraju isprogramirati zlog duha koji će se nasumično kretati po labirintu. Ako glavni lik dotakne duha, igra je gotova.</p> <p>Cilj: Ovom aktivnošću učenici će preispitati svoje znanje o kretanju unutar labirinta, koristeći blok osjetilo boja koje su naučili u</p>



	prethodnim aktivnostima. Upoznat će se s konceptom kloniranja objekta s ograničenjima položaja, te stvaranjem jednostavnih likova, koji nisu igrači, sa vlastitim nasumičnim kretanjem.
Trajanje	90 minuta
Strategija i metode učenja i poučavanja	Aktivno učenje Suradničko učenje Rješavanje problema
Oblici poučavanja	Frontalni rad Rad u paru Individualni rad Grupni rad (svi učenici)
Razrada aktivnosti	<p>Učenicima se prezentira igra koju trebaju izraditi: Igrač skuplja nasumično postavljene zvijezde dok ga progoni crveni duh. Ako se igrač i duh sudare, igra je gotova. Ako igrač sakupi 20 zvijezda, on pobjeđuje.</p> <p>[Korak 1]</p> <p>Upućujemo učenike da dizajniraju labirint, gdje je područje unutar kojeg se igrač može kretati jedne boje (npr. plave), a zidovi koji ograničavaju kretanje igrača druge boje (npr. crne). Kako bi uštedili na vremenu možemo unaprijed pripremiti pozadinsku sliku labirinta.</p>  <p>[Korak 2]</p> <p>Učenici trebaju nacrtati lik <i>pacman</i> i crvenog duha. Kao zvijezdu mogu nacrtati krug unutar Snap!-a:</p>  <p>[Korak 3]</p> <p>Kako bi napravili kretanje pacman-a, možemo koristiti različite mogućnosti. Primjer ispod je jedan od njih. U njemu koristimo operacije upravljanja za određivanje događaja, tj. tipke koja je pritisnuta: lijevo, desno, gore ili dolje. Nakon svakog od događaja, moramo testirati dodiruje li lik boju područja unutar kojeg mu je dozvoljeno kretanje. U tom slučaju, najprije se okreće u smjeru</p>



kretanja i pomakne se. Ukoliko dotakne boju zida, mora se pomaknuti unatrag jer bi u protivnom zapeo u zidu zbog prvog slučaja.



[Korak 4]

Sljedeći korak je isprogramirati zvijezde. Zvijezde će biti sve jednake, ali će ih biti mnogo. U ovom slučaju, umjesto izrade više jednakih objekta (u našem slučaju 20), bolje je napraviti jedan objekt pa zatim izraditi njegove klonove. Na početku igre prvi klon će se pojaviti nasumično unutar labirinta. Kada ga zatim igrač pokupi, nestat će i novi će biti napravljen na drugoj nasumičnoj lokaciji. Kako bi izradili prvog klonu na samom početku igre, stavljamo sljedeći kod na scenu skripte.



Kako bi sakrili originalni objekt i prikazali samo klonove, moramo ovo napraviti na samom početku programa.

Kako bi pronašli odgovarajuću nasumičnu lokaciju moramo postaviti određene uvijete. Ako je zvijezda kreirana na zidu, igrač je ne može doseći, što znači da je ne smijemo postaviti na zid. Strategija za postavljanje je sljedeća.

1. Moramo pronaći nasumičnu x,y poziciju klena zvijezde. x i y



- koordinata su unutar intervala [-140, 140]. Stoga izabiremo nasumičan broj iz intervala za obje koordinate.
2. Zatim provjerimo dodiruje li klon boju zida. U tom slučaju lokacija nije dozvoljena.
 3. Ako je lokacija dozvoljena, moramo prikazati klona (prisjetite se, originalni objekt je skriven i klon bi isto bio skriven kada ne bi koristili blok za prikaz objekta) i u beskonačnoj petlji provjeriti dolazi li do preklapanja sa igračem.
 4. Ako lokacija nije dozvoljena, izradimo novog klona (nadajući se da će novo odabrani nasumični brojevi biti takvi da je klon postavljen na dozvoljenu lokaciju) i brišemo postojećeg.
 5. Kako bi izbrojali prikupljene klone moramo dati informaciju brojaču zvijezda koji mora biti definiran izvan klona, npr. na igraču. To može biti napravljeno na način da se prenese poruka da je došlo do preklapanja. Zatim je možemo izbrisati.



[Korak 5]

Nadalje programiramo duha. Duh se mora kretati nasumično kroz labirint i mora promijeniti smjer kada dođe do zida. Kako bi napravili njegovo kretanje nasumičnim, želimo da se kreće u nasumičnom smjeru nakon što dođe do zida. U Snap!-u smjer kretanja je izražen stupnjevima:

1. 0 stupnjeva – GORE
2. 180 stupnjeva – DOLE
3. 90 stupnjeva – DESNO
4. 270 stupnjeva – LIJEVO

Drugim riječima, ako slučajno odabrani broj od 0 do 3 pomnožimo sa 90 dobit ćemo nasumični smjer kretanja.

Duh se mora kretati dok se ne sudari sa pacman-om. Tada je igra gotova.



[Korak 6]

Sada trebamo napisati program kada igrač pobjeđuje u igri. To će se dogoditi kada igrač sakupi 20 zvijezda. Brojač zvijezda imamo unutar pacman skripte. Na samom početku postavimo inicijalnu vrijednost na 0, i zatim joj povećavamo vrijednost za 1 svaki puta kada klon pošalje poruku da je igrač sakupio zvijezdu. Ako brojač dođe do 20, pacman pobijeđuje i moramo zaustaviti igru.



Alati i materijali za nastavnike	Igra izrađena alatom Snap!: https://snap.berkeley.edu/project?user=zapusek&project=pacman_clone Lajovic, S. (2010). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena. Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.
Alati i materijali za učenike	Predložak izrađen alatom Snap!: https://snap.berkeley.edu/project?user=zapusek&project=pacman_template

REFERENCE

Lajovic, S. (2011). Scratch. *Nauči se programirati in postani računalniški maček*. Ljubljana: Pasadena.

Rugelj, J. (2019). Game design based learning of programming.

Vorderman, C. (2017). *Računalniško programiranje za otroke*. Ljubljana: MK.

Дурева Д., М. Касева, Г. Тупаров, Компютърно моделиране, 4. клас, Просвета, 2018,

София (Dureva, D., M. Kaseva, G. Tuparov, Kompyutarno modelirane, 4. klas, Prosveta, 2019, Sofia)