# O3 – Izobraževalne vsebine za učitelje

ZBIRKA UČNIH SCENARIJEV ZA UČENJE PROGRAMIRANJA NA OSNOVI OBLIKOVANJA IGER





#### Podatki o dokumentu

**Rezultat**: O3/A1 – Zbirka učnih scenarijev za učenje programiranja na osnovi oblikovanja iger

Intelektualni rezultat: O3 – Izobraževalne vsebine za učitelje

Odgovorni partner: South-West University "Neofit Rilski" (Bolgarija)

Sodelujoči partnerji: Univerza v Ljubljani (Slovenija), Univerza na Reki (Hrvaška)

#### Izjava o omejitvi odgovornosti

Projekt CODING4GIRLS je financiran iz programa Erasmus+ Evropske unije.

Za vsebino tega dokumenta so odgovorni izključno avtorji. Vsebina ne odraža nujno mnenja Evropskih skupnosti. Evropska komisija ne prevzema odgovornosti za kakršnokoli uporabo informacij, ki jih ta dokument vsebuje.

Coding4Girls, 2018-2020



Creative Commons Attribution-ShareAlike 4.0 International Public License (<u>CC BY-SA 4.0</u>)





1.	UVOD	1
2.	UČNI SCENARIJI	5
	OSNOVNI UČNI SCENARIJI	ô
	Učni scenarij 1 – Uvod v okolje Snap!	ô
	Učni scenarij 2 – Lik oživi1	1
	Učni scenarij 3 – Premikanje po odru1	5
	Učni scenarij 4 – Menjava obleke in obrat2	1
	Učni scenarij 5 – Zvoki na kmetiji28	3
	Učni scenarij 6 – Kameleon na počitnicah36	ô
	Učni scenarij 7 – Pomagaj princu in princeski najti svoje živali46	ô
	Učni scenarij 8 – Risanje s kredo53	3
	Učni scenarij 9 – Pobiranje smeti in čiščenje parka63	3
	Učni scenarij 10 – Nahrani mačke7	1
	Učni scenarij 11 – Mačje zavetišče79	9
	NAPREDNEJŠI UČNI SCENARIJI88	3
	Učni scenarij 12 – Lovljenje zdrave hrane88	3
	Učni scenarij 13 – Sestavi zgodbo96	ô
	Učni scenarij 14 – Onesnažen zrak106	ô
	Učni scenarij 15 – Ulovi miš117	7
	Učni scenarij 16 – Kupovanje hrane za piknik12	5
	Učni scenarij 17 – Računanje133	3
	Učni scenarij 18 – Recikliranje14	1
	Učni scenarij 19.1 – Zaigraj na klavir 1144	1
	Učni scenarij 19.2 – Zaigraj na klavir 2152	1
	Učni scenarij 20 – Test158	3
	Učni scenarij 21 – Enostavni PACMAN162	1
3.	VIRI	3





Vodilni psiholog prejšnjega stoletja je identificiral igro kot eno najpomembnejših aktivnosti za razvoj pomembnih življenjskih veščin, ne glede na starost ali stopnjo razvoja. Otrok se skozi igro hitro prilagaja in sprejema nove okoliščine. Med igro odkriva osnovne pojme iz resničnega sveta in na ta način se oblikujejo prvi odnosi med njima.

Dandanes se igre najpogosteje uporabljajo v zgodnjih fazah otrokovega razvoja doma in v vrtcu, v šoli pa učenje še vedno prepogosto temelji na tradicionalnem prenosu znanja z učiteljem v glavni vlogi in pasivnimi učenci. Po drugi strani pa teorije učenja, razvite v prejšnjem stoletju, spodbujajo k novim pristopom učenja in poučevanja, ki so usmerjeni na učenca, temeljijo na reševanju problemov, so usmerjeni k doseganju višjih taksonomskih ravni, predstavljajo motivacijo za učence in so pogosto podprti z informacijsko-komunikacijsko tehnologijo (IKT).

S pristopom CODING4GIRLS bomo učence spodbudil k sodelovanju v programerskih aktivnostih, kjer bodo najprej reševali lažje primere, kasneje pa bodo prešli na čedalje zahtevnejše izzive, namenjene predvsem uspešnejšim učencem (*low entry - high ceiling approach*). Učence bomo spodbudili k dokončevanju delnih rešitev ali k ustvarjanju popolnoma novih. Aktivnosti si sledijo po težavnosti: od osnovnih z enim programerskim konceptom do zahtevnejši z več programerskimi koncepti. Pripravljene aktivnosti v vizualnem programskem jeziku Snap! smo poskušali povezati z resničnimi problemi ter upoštevati preference deklet pri igranju iger.

Pripravljeni učni scenariji vsebujejo vse informacije, ki jih potrebujejo učitelji za uporabo pristopa učenja programiranja z izdelavo iger in s tem gradnje programerskih veščin za deklice in dečke. V učnih scenarijih so na voljo:

- pričakovano predznanje,
- splošni učni cilji,
- koncepti, ki jih aktivnost pokriva,
- specifični učni cilji,
- pričakovani rezultati,
- navodila po korakih za izvedbo Coding4Girls pristopa učenja z izdelavo iger,
- vprašanja za spodbujanje razprave med učenci v razredu.

Pripravili smo več kot 20 učnih scenarijev. Učitelji lahko uporabijo scenarije v predlaganem zaporedju ali pa jih prilagodijo svojim željam in potrebam. Scenariji opisujejo splošno funkcionalnost predlagane izobraževalne igre, načine interakcije z uporabniki, povratne informacije in opise učnih aktivnosti v predlagani igri.

Učni scenariji so na voljo v angleškem, bolgarskem, grškem, hrvaškem, italijanskem, portugalskem, turškem in slovenskem jeziku.



# 2. UČNI SCENARIJI

Pripravljeni učni scenariji so razvrščeni v dva dela: osnovni učni scenariji pokrivajo en programerski koncept, naprednejši pa več. Spodnja tabela prikazuje predlagan vrstni red aktivnosti.

1	Uvod v okolje Snap!	UL
2	Lik oživi	UL
3	Premikanje po odru	UL
4	Menjava obleke in obrat	UL
5	Zvoki na kmetiji	UL
6	Kameleon na počitnicah	UL
7	Pomagaj princu in princeski najti svoje živali	UL
8	Risanje s kredo	UL
9	Pobiranje smeti in čiščenje parka	UL
10	Nahrani mucke	UL
11	Mačje zavetišče	UL
NAPR	EDNEJŠI UČNI SCENARIJI	
12	Lovljenje zdrave hrane	UL
13	Sestavi zgodbo	SWU
14	Onesnažen zrak	UNIRI
15	Ulovi miš	UL
16	Kupovanje hrane za piknik	UL
17	Računanje	SWU
18	Recikliranje	SWU
19.1	Zaigraj na klavir 1	SWU
19.2	Zaigraj na klavir 2	UNIRI
20	Test	SWU
21	Enostavni Pacman	UL





OSNOVNI UČNI SCENARIJI

# Učni scenarij 1 – Uvod v okolje Snap!

Naslov učnega	Uvod v okolje Snap!
scenarija	
Pričakovano	/
programersko	
predznanje	
Učni cilji	Splošni učni cilji:
	<ul> <li>seznanitev z vizualnim programskim okoljem Snap!</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>Učenec zna dodati nov lik</li> <li>Učenec zna liku dodati novo obleko in je uroditi</li> </ul>
	<ul> <li>Učenec zna nastaviti središče vrtenja lika tako, da zgleda</li> </ul>
	njegovo vrtenje čim bolj naravno
Cilji, naloge in kratek	Učenec doda nov lik, ki mu doda obleko. Obleko uredi in izbriše enega
opis aktivnosti	od njih. Učenec ustvari novo ozadje in izbriše neželjenjega
	Cilji: Ob koncu učne ure bo učenec narisal svoj izbrani lik in okolje v
	katerem živi, resnično ali izmišljeno, da bi ga v nadaljevanju lahko
	uporabil v igri. Študije so pokazale, da je na ciljno skupino
	ustvarjanje lastnih likov deluje motivacijsko.
Trajanje aktivnosti	45 minut
Učne strategije in	Demonstracija
metode	Individualno delo
Učne oblike	Frontalno delo
	Individualno delo
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	
	Ob koncu učne ure bodo učenci narisali svoj najljubši lik in okolje, v





katerem živi, resnično ali izmišljeno, da bi ga kasneje lahko uporabili v igri.

[1. korak]

Pokažite učencem, kje lahko najdejo Snap!

(<u>https://snap.berkeley.edu/</u>). Pokažite jim različne dele uporabniškega vmesnika: predel, kjer najdejo različne delčke kode, predel, kjer sestavljajo svoj program,/ spreminjajo obleke/dodajo zvok, oder, na katerem je lik, seznam likov.





Nov lik lahko ustvariš s klikom na enega od treh gumbov:



Ker boste narisali svoj lik, kliknete na čopič in odprl se bo urejevalnik slik, ki je podoben Risarju.





Naloga za učence: Narišite svoj prvi lik. Na voljo imate 10 minut. Ko narišejo lik, morajo preveriti, da je središče vrtenja njihovega lika

na pravem mestu. Za to naj uporabijo orodje

Naloga za učence: določite točko, okoli katere se bo vrtel vaš lik.

### [3. korak]

Za urejanje lika izberite zavihek obleke, ki je viden le, če je trenutno izbran lik. Po desnem kliku na obleko, ki jo želite urediti, se odpre spustni seznam, v katerem izberete uredi. Obleko lahko v istem seznamu tudi podvojite ali izbrišete.



# [4. korak]

Za uvoz že predpripravljene obleke, kliknite na ikono, ki izgleda kot list papirja, in izberite Obleke...







Tudi tokrat velja, da se ta možnost pojavi le, če je trenutno izbran lik. Naloga za učence: izberite si eno obleko in jo dodajte svojemu liku. [5. korak]

Zdaj ko imate svoje like, jim boste dodali še ozadje. Najprej pod odrom namesto na lik, kliknite na Oder. Da bi dodali novo ozadje, izberite zavihek Backgrounds:



Naloga za učence: nariši svoje ozadje.

Naloga za učence: dodaj še eno od vnaprej pripravljenih ozadij, da boš imel dve. Najdeš jih na podoben način kot prej obleke za lik

Naloga za učence: poišči način, kako lahko urediš ozadje in kako ga izbrišeš.

(Refleksija in evalvacija)

Ali so učenci uspeli narisati svoj lik in ozadje? So imeli kakšne težave? Kako so jih reševali?

Učni pripomočki,	https://snap.berkeley.edu/
sredstva za učitelja	
Učni pripomočki za	Navodila za učenca (C4G1_NavodilaZaUcenca.docx)
učenca	









# Učni scenarij 2 – Lik oživi

Naslov učnega	Lik oživi		
scenarija			
Pričakovano	/		
programersko			
predznanje			
Učni cilji	Splošni učni cilji:		
	<ul> <li>Učenec ve, kje v programskem vmesniku lahko najde delčke in kako jih poveže v zaporedje</li> <li>Učenec zna sestaviti zaporedje delčkov za premikanje lika</li> </ul>		
	<ul> <li>Učenec zna uporabiti delček reci</li> </ul>		
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:		
<ul> <li>Sestaviti ustrezno zaporedje delčkov</li> </ul>			
Cilji, naloge in kratek	Učenci izvejo, kje so shranjeni programski delčki in kako najdejo		
opis aktivnosti	ustreznega, katere kategorije delčkov poznamo in kako jih		
	povežemo v smiselno zaporedje		
Trajanje aktivnosti	45 minut		
Učne strategije in	Demonstracija		
metode	Individualno delo		
Učne oblike	Frontalno delo		
	Individualno delo		
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)		
procesa	To uro boste spoznali, kako z bloki likom naročimo naj se premikajo po		
	odru in govorijo.		
	Učencem lahko pokažete primer programa, ki ga bodo znali ob koncu		
	ure narediti.		
	[1. korak]		
	Najprej na zaslonu poiščite, kje najdete delčke kode, ki jih lahko		
	uporabite za sestavljanje svojega programa. Kje se nahajajo?		
	Na levi strani, nad delčki, lahko vidite, da imate različne kategorije		
	delčkov:		





Sofinancira program Evropske unije

premikanje, izgled, zvok, svinčnik, krmiljenje, zaznavanje, operatorji in

spremenljivke.

Snap!	
Premikanje	Krmiljenje
<b>Izgled</b>	Zaznavanje
Zvok	Operatorji
Svinčnik	Spremenlijvk

Ti delčki so obarvani z različnimi barvami in s tem povezani v skupine,

na primer vsi delčki za premikanje so modre barve.

Naloga za učence: Najprej poiščite delček za premikanje in dvakrat kliknite nanj. Kaj se je zgodilo?

[2. korak]

Da se bo lik premikal naprej, moramo najprej povleči in izpustiti

premakni se (10) korakov z levega dela okna na del, ki je delček

namenjen sestavljanju skript, to je delčkov programa:



premakni se (10) korakov Če sedaj dvakrat kliknete na delček , se bo vaš lik premaknil za 10 korakov.

[3. korak]

Program v Snap!-u se običajno začnejo s klikom na zeleno zastavico. Naloga za učence: poglejte v različne kategorije in poiščite delček, s katerim se bo program začel izvajati, ko kliknemo na zeleno zastavico. Povlecite ga med skripte.





Sofinancira program Evropske unije Erasmus+







	[7. korak]		
	Naloga za učence: Odprite program, ki ste ga ustvarili prejšnjo uro.		
	Svoj lik povlecite na levi del odra in sestavite program, ki poskrbi, da		
	se vaš lik premika z izbrane točke na levi proti desni strani odra. Po		
	vsakem premiku naj nekaj pove. Narediti more več kot le en premik.		
	Program večkrat poženite. Ali lik po vsakič vedno konča na isti točki?		
	Poiščite delček, ki poskrbi, da se bo lik vedno najprej vrnil na začetno		
	točko in ne bo pobegnil z zaslona.		
	Namig za učitelja. Če lik pobegne z zaslona, ga pokličite nazaj tako, da		
	ga poiščete med liki in z desnim miškinim gumbom kliknete nanj. V		
	seznamu izberite prikaži.		
	Delček, ki ste ga iskali je pojdi na x: 190) y: 85. Za določitev		
	pravega x in y, lahko najprej premaknete svoj lik na željeno točko in		
	pod delčki v kategoriji Premikanje odkljukate okence pri položaj x in		
	položaj y, da se vam izpiše trenutni položaj lika. Ti dve vrednosti vpišeš		
	v prostorčka za x in y.		
	(Refleksija in vrednotenje)		
	Kolikokrat je moral vaš lik ponoviti zaporedje premikanja in		
	govorjenja, da je opravil nalogo? Ste vsi v razredu naredili enako		
	število ponovitev? Zakaj?		
Učni pripomočki,	Primer programa:		
sredstva za učitelja	https://snap.berkeley.edu/project?user=mateja&project=C4G2 Kuza		
	<u>GreDomov</u>		
Učni pripomočki za	a Učenci, ki prejšnjo uro niso izdelali svoje lika in ozadja, lahko		
učenca	uporabijo:		
	<ul> <li>Predloga aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=mateja&amp;project=C4G</u> <u>2 KuzaGreDomov tmp</u></li> </ul>		
	Navodila za učence, ki delajo samostojno:		
	<ul> <li>Navodila za učenca (C4G2_NavodilaZaUcenca.docx) – navodila</li> </ul>		





za učence so namenjena učencem, ki delajo samostojno.

# Učni scenarij 3 – Premikanje po odru

Naslov učnega	Premikanje po odru		
scenarija			
Pričakovano	/		
programersko			
predznanje			
Učni cilji	Splošni učni cilji:		
	<ul> <li>Sestaviti smiselno zaporedje delčkov</li> </ul>		
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:		
	<ul> <li>Učenec ve, kako postaviti lik na določeno mesto na odru</li> <li>Učenec zna premakniti li v smeri x in smeri y</li> <li>Učenec zna uporabiti zanko ponovikrat</li> </ul>		
	<ul> <li>Učenec ve, da je smer gibanja lika pri delčku pojdi korakov relativna glede na smer, v katero je lik obrnjen</li> </ul>		
Cilji, naloge in kratek	<b>Kratek opis:</b> Učenec spozna, kako premikati lik po odru v x in y smeri.		
opis aktivnosti Učenec sestavi preprost program za rešitev naloge. Na			
	obračanja lika v različne smeri in kako to vpliva na delček pojdi		
	korakov.		
	Naloge: ustvari program, ki bo premaknil lik v smeri x, ustvari		
	program, ki bo premaknil lik v smeri y, ustvari program, ki bo združil		
	gibanje v x in y smeri.		
	Cilji: razlikovanje med premiknjem po odru v x in v y smeri ter		
	uporaba zanke ponovi krat		
Trajanje aktivnosti	45 minut		
Učne strategije in	Demonstracija		
metode	Individualno delo		
Učne oblike Frontalno delo			
	Individualno delo		





Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	Različnim živalim boste pomagali, da dosežejo svoj cilj. Za to jim boste
	mroali podati jasna navodila o tem, kako se morajo premikati po odru.
	[1. naloga]
	Odprite nalogo Ujemi žogo
	(https://snap.berkeley.edu/snap/snap.html#present:Username=
	mateja&ProjectName=C4G3 Ujemi zogo).
	Naredite animacijo premikanja psa z levega dela odra proti žogi. Pri
	tem uporabite delčka spremeni x za 🕥 in <mark>čakaj 🔿 sekund.</mark>
	Možna rešitev naloge:
	ko Hiknemp na pojdi na x: fij yr 420 čakoj (    sekund) spremeni x za 20 čakoj (    sekund)
	Kot vidite se pri premikanju levo in desno po odru spreminja vrednost
	x. Ko je x 0, je vaš lik na sredini odra. Vse kar se nahaja levo od njega





mora imeti pred številko oznako -, dlje stran kot je od sredine, večja je številka poleg znaka -. Desno od sredine so vrednost x večje od 0 in pred njih ne pišemo nobenega predznaka. Namig: Če učenci poznajo decimalna števila, lahko čas čakanja zmanjšamo na 0.1 sekunde. Če učenci že poznajo negativna števila, lahko to uporabimo pri razlagi negativnih vrednosti x. Če učenci že poznajo koordinatni sistem, razlago vrednosti x navežemo na to temo. [2. naloga] Odprite nalogo Plezanje opice () in sestavite naredite animacijo opice, ki pleza na palmo, da bi dosegla banane. Pri tem uporabite delčka in cakaj O sekund. . Pazite, da bo opica vedno spremeni y za 🔵 začela s plezanjem z istega mesta. Dodatna naloga: Ko opica pride do banan, naj spleza nazaj na začetno mesto. Možna rešitev osnovne naloge: -120 neni y za 10. 1 teni y za 10 10 meni y za





Kot lahko vidite, se vrednost y spreminja, ko se lik premika gor in dol po odru. Če je vrednost y 0, je lik na sredini odra. Vse kar je višje od sredine odra, ima vrednost y večjo od 0. Če se vaš lik nahaja nižje od sredine odra, je to podobno kot pri potapljanju: da si pod vodno gladino poveš s tem, da pred številko napišeš -, s številko pa poveš, koliko »metrov« pod vodno gladino si. V našem primeru pa ti številka za znakom - pove, koliko korakov pod sredino odra je lik. Če želiš, da se opica s palme vrne nazaj na začetno mesto, uporabite

### spremeni y za -10

Namig: Učencih, ki že poznajo decimalna števila, lahko za boljšo animacijo pri čakanju uporabijo decimalno število (na primer 0.1). Če učenci že poznajo koordinatni sistem, razlago premikanja v y smeri navežite na to snov.

### [3. naloga]

V obeh nalogah ste izmenično uporabljali dva enaka delčka. Kolikokrat ste morali **ponoviti** ta dva delčka z enakimi podatki? Obstaja krajši način za pisanje kode, s katerim računalniku poveš, kolikokrat naj ponovi enake ukaze. Delček, s katerim računalniku povemo, katere kaze naj ponovi in kolikokrat naj to stori, je zanka

ponovi 🔵 krat

Uporabite jo lahko, kadar želite enake ukaze večkrat zapored ponoviti v istem vrstnem redu. Poskusite svoja programa, ki ste ju napisali danes, spremeniti tako, da boste uporabili zanko ponovi \_\_\_ krat. Ukazi, za katere želite, da se ponovijo, morate pripeti znotraj delčka in napisati, kolikokrat naj se to zaporedje ukazov ponovi.

















# Učni scenarij 4 – Menjava obleke in obrat

Naslov učnega	Menjava obleke in obrat		
scenarija			
Pričakovano	Premikanje po odru z ukazi pojdi _ korakov, spremeni x za _ in		
programersko	spremeni y za _		
predznanje	Poznavanje zanke ponovi krat		
Učni cilji	Splošni učni cilji:		
	<ul> <li>Sestavljanje smiselnega zaporedja delčkov</li> </ul>		
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:		
	<ul> <li>Učenci znajo liku zamenjati obleko in tako narediti animacijo</li> <li>Učenci znajo nastaviti različne načine obračanja lika in lik obrniti</li> </ul>		
Cilji, naloge in kratek	Kratek opis: Učenci spoznajo, kako naredijo animacijo lika s		
opis aktivnosti	spreminjanjem njegove obleke. Naučijo se, kako določiti praviler		
	način obračanja lika glede na njegove lastnosti.		
	Naloge: Ustvari programe, v katerih lik menjuje obleko. V vsakem		
	programu nastavi ustrezen način obračanja lika.		
	Cilji: znati liku zamenjati obleko in nastaviti ustrezen način obračanja		
	lika.		
Trajanje aktivnosti	45 minut		
Učne strategije in	Demonstracija		
metode	Individualno delo		
Učne oblike	Frontalno delo		
	Individualno delo		
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)		
procesa	V tej uri se boste naučili, kako narediti tako animacijo lika, da bo		
	izgledalo kot da lik pleše, hodi		





#### [1. naloga]

Odprite prazen projekt in kliknite na ikono, ki izgleda kot bel list papirja v levem zgornjem kotu in kliknite na Obleke. Izberite obleko ballerina a in kliknite na Import. Enako naredite še z oblekami ballerina b, ballerina c in ballerina d. Nato kliknite Prekliči. V zavihku Obleke imate sedaj 4 obleke svojega lika. Ime lika lahko spremenite v Balerina. To naredite tako, da dvakrat kliknete na besedilo Objekt in vpišete poljubno ime lika.



Vrnite se na zavihek Skripte in napišite kodo, v kateri bo lik začel z obema nogama na tleh, nato pa bo plesal tako, da bo 15x spremenil svojo obleko. Pri tem boš uporabil delčka našlednja obleka. Na koncu naj lik zamenja obleko tako, da bo svoj ples končala z obema nogama na tleh. Poskrbi, da bodo vidni vsi gibi v plesu in ne pozabi na začetek programa dodati delčka, ki programu sporoči, kdaj naj se začne izvajati!





#### Primer rešitve:



Naša balerina noče biti ves čas na istem mestu, zato se vsakič, ko zamenja obleko, še malo premakne. Dopolnite svojo kodo in si shranite program.

#### Možna rešitev:



# [2. naloga]

Odprite nov prazen projekt in uvozite vse obleke za lik avery walking. Dodajte ozadje, po katerem se bo Avery lahko sprehajala. Ustvarite animacijo, v kateri se Avery sprehaja od leve proti desni strani odra. Poskusite ugotoviti, kako Avery animirati tako, da bo izgledalo kot da naslednji korak kot v realnem življenju nadaljuje tam, kjer je prejšnjega končala. Ne pozabite shraniti svojega programa.

Možna rešitev:





Sofinancira program Evropske unije Erasmus+



### [3. naloga]

Doslej smo sestavljali le programe, pri katerih se lik ni obračal. V tej nalogi pa imamo miško, ki jo moramo po cikcakasti poti pripeljati do



sira, zato jo bomo morali na poti tudi obrniti v ustrezno smer. Za to lahko uporabimo delček obrni se v smeri \_, kjer miši povemo, v katero smer naj gleda: right – desno left – levo

up – gor

down – dol

random – naključno

Druga možnost pa je, da miši povemo, za kakšen kot naj se obrne. To lahko naredimo z dvema ukazoma: obrni se v smeri urinega kazalca obrni se (15 stopini) ali obrni se v nasprotni smeri od urinega kazalca obrni se (15 stopini). Cel krog meri 360°. Če želite miš obrniti v nasprotno smer, jo obrnete za 180°; če jo želite obrniti desno, jo obrnite za 90° v smeri urinega kazalca; če jo želite obrniti levo, jo obrnite za 90° v nasprotni smeri od urinega kazalca. [Naloga] Odprite predlogo (https://snap.berkeley.edu/snap/snap.html#present:Username=





Sofinancira program Evropske unije Erasmus+

<u>mateja&ProjectName=C4G4 Najdi sir</u>) in napišite program, s pomočjo katerega se bo miš sprehodila do sira. Pri tem se lahko premika le po zeleni poti. Usmerite miš v pravo smer in za premikanje uporabite delček pojdi \_ korakov. Da boste videli, kako se mi š počasi premika, uporabite ukaz čakaj 1 sekund.

### Možna rešitev:



Poskusite napisati isti program še tako, da uporabite ukaza obrni se \_ stopinj.

# Možna rešitev:



[4. naloga]

Kot ste lahko opazili, se miš med premikanjem obrača v različne smeri.





Včasih ne želimo, da se lik obrača v vse smeri – na primer ne želimo, da bi Avery hodila po glavi, ampak želimo, da se obrača le levo in desno. Zato da se bo vaš lik obračal na tak način kot si želite, moramo določiti ustrezen način obračanja. To naredite teko, da levo od lika izberete eno od naslednjih možnosti:

	Mis
	viečijiv
	Skripte Obleke Zvoki
	<ul> <li>a. Krožna puščica pomeni prosto vrtenje in lik se lahko obrača v vse smeri (kot tvoja miš)</li> <li>b. Puščica levo-desno pomeni, da se lahko lik obrača le levo ali desno (to lahko uporabiš pri psu, da ne hodi »po glavi«)</li> <li>c. Puščica desno pa pomeni, da se lik ne vrti in bo vedno gledal v isto smer (to lahko uporabiš pri opici).</li> </ul>
	V programih, ki ste jih sestavili prejšnjo uro (s psom in opico) nastavite
	da se bo pes lahko sprehodil do žoge, se obrnil za 180° in se vrnil na
	izhodišče, ne da bi se obrnil na glavo. Podobno naredite še z Avery.
	[Refleksija in preverjanje znanja]
	Pri katerih od programov, ki ste jih sestavili do sedaj je smiselno
	uporabiti prosto vrtenje lika, pri katerih obračanje levo-desno in pri
	katerih ne vrti?
	Se vam zdi lažje obračanje likov v določeno smer ali za določeno
	število stopinj? Ali je vedno vseeno, kateri način obračanja
	uporabimo?
lčni pripomočki,	Primer celotne aktivnosti v Snap!-u:
redstva za učitelja	Balerina:
	https://snap.berkeley.edu/project?user=mateja&
	project=C4G4 Balerina
	<ul> <li>Avery: <u>https://snap.berkeley.edu/project?user=mateja&amp;</u></li> </ul>





	•	project=C4G4 Avery Najdi sir: https://snap.berkeley.edu/project?user=mateja& project=C4G4 Najdi%20sir%20resitev
Učni pripomočki za	•	Predloga aktivnosti v Snap!-u za Najdi sir:
učenca		https://snap.berkeley.edu/snap/snap.html#present:Username
		=mateja&ProjectName=C4G4 Najdi sir
	•	Navodila za učenca (C4G4_NavodilaZaUcenca.docx)





# Učni scenarij 5 – Zvoki na kmetiji

Naslov učnega	Zvoki na kmetiji
scenarija	
Pričakovano	• Dodajanje ozadja;
programersko	<ul> <li>dodajanje novega lika;</li> </ul>
predznanje	• govorjenje lika.
Učni cilji	Splošni učni cilji:
	<ul> <li>dodajanje zvoka iz knjižnice orodja Snap!;</li> <li>dodajanje zvoka iz računalnika;</li> <li>snemanje zvoka;</li> <li>predvajanje zvoka ob pritisku na tipko.</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>učenec doda zvok iz knjižnice orodja Snap! in ga doda v kodo tako, da se začne predvajati ob pristiku na določeno tipko;</li> <li>učenec uvozi zvok iz računalnika in ga doda v kodo tako, da se začne predvajati ob pristisku na določeno tipko;</li> <li>učenec posnema zvok in ga doda v kodo tako, da se začne predvajati ob pritistku na določeno tipko.</li> </ul>
Cilji, naloge in kratek	Kratek opis: Učenci sestavijo enostavno igro, v kateri igralec s
opis aktivnosti	pritiskom na določeno tipko spozna oglašanje živali na kmetiji.
	Naloga: Sprva morajo učenci izbrati ozadje igre. Nato dodajo
	gospodarico kmetije, ki pove navodila igre: »Če želiš slišati psa, klikni
	na tipko P!«. Podobna navodila pove tudi za ostale živali na kmetiji.
	Učenec po navodilih gospodarice sestavi kodo igre.
	<b>Cilj:</b> Učenec spozna kako dodati in uporabiti zvok v orodju Snap!.
	Pri tem se bo naučil tudi kako uporabiti zvočni blok
	( <i>predvajaj_zvok_[zvok]</i> ) in kontrolni blok
	(ko_pritisnemo_na_tipko_[tipka]).
Trajanje aktivnosti	45 minut
Učne strategije in	Aktivni pouk, učenje z ustvarjanje iger
metode	





Učne oblike	Frontalna oblika
	Individualno delo
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	Motivacija-uvod
	Za uvodno motivacijo učenci igrajo primer že ustvarjene igre Zvoki na
	kmetiji, pri čemer ne vidijo kodo igre. Učenci vidijo cilj učne ure in so
	motivirani za ustvarjanje svoje igre.
	Constantiant Table The The Table of The Table of The Table of The Table of
	[1. korak]
	Učenci morajo najprej izbrat ozadje svoje igre. Ozadje mora vsebovati
	različne živali na kmetiji. Kot učitelji lahko izbiramo med naslednjimi
	možnostmi:
	<ol> <li>učencem ponudimo možnost, da sami narišejo ozadje (v tem primeru je priporočljivo, da risanje ozadja časovno omejimoa);</li> <li>učenci na spletu poiščejo sliko kmetije, ki je prosto dostopna in ima dovoljenje za uporabo ali pa jim ponudimo spletne strani, ki vsebujejo omenjene slike;</li> <li>učitelj učencem pripravi predlogo programa, ki že vsebuje ozadje (v tem primeru imajo vsi učenci enako ozadje).</li> </ol>
	Učenci so dodajanje ozadja spozanli v predhodnih urah, zato to
	naredijo individualno.





Sofinancira program Evropske unije Erasmus+

#### [2. korak]

V naslednjem koraku učenci dodajo gospodarico kmetije. Tudi v tem

primeru se učitelj odloči za eno izmed možnosti:

- 1. učenci sami narišejo gospodarico;
- 2. učenci uvozijo prosto dostopno sliko;
- 3. učitelji pripravi predlogo programa, ki vsebuje gospodarico kmetije.



### [3. korak]

Ko je dodana gospodarica kmetije, ji lahko dodamo kodo tako, da bo na začetku igre povedala navodila. Učenci pri tem uporabijo bloke kot so *lzgled/reci\_[stavek]\_za\_[x] sekund* in *počakaj\_[x]\_sekund*. Učenci so te bloke spoznali v predhodnih učnih urah, zato lahko ta korak opravijo individualno.







V prvih 3. korakih so učenci ponovili snov iz prejšnjih učnih ur. V nadaljevanju pa bodo spoznali kako v igro dodajati zvok. Za začetek jim učitelj frontalno pokaže tri načine dodajanja zvoka:

- 1. dodajanje zvoka iz knjižnice orodja Snap!;
- 2. uvoz zvoka iz računalnika na način polveči in spusti;
- 3. snemanje zvoka v orodju Snap!.

Vsi zvoki se uvozijo v zavihek *Zvoki*. Ko se učenci spoznajo z načini dodajanja zvoka, nadaljujejo z ustvarjanjem igre individualno s podporo učitelja.

### [4. korak]

Učenci najprej dodajo zvok psa iz knjižnce orodja Snap!. Knjižnico zvokov najdemo s klikom na ikono belega lista, kjer se na koncu spustnega seznama ponudi možnost *Zvoki*.

Premikanie	Opis projekta	
Izaled	Nov	^N I farmer
Tuck	Odpri	^0 Iv
Svinčnik	Shrani Shrani kot	Zvoki
Preklopi na obi naslednja oblek Št.obleke	Uvozi Izvozi projekt (in a new window) Izvozi projekt kot navadno besedil Povzetek izvoza Export summary with drop-shadow Export all scripts as pic	lo NS
reci Halo! za 2	Knjižnice Obleke	Select a sound from the media library
reci Halo!	Zvoki	

Odpre se knjižnica zvokov, v kateri učenci poiščejo zvok psa, ga označijo in za uvoz v program kliknejo na gumb *Import*.







Ko je zvok psa dodan pod zavihek *Zvoki*, ga je potrebno vključiti v kodo igre. Učenci morajo dodati kodo tako, da bo pes zalajal, ko bo igralec pritisnil na tipko »D«. Pri tem si pomagajo z blokoma *ko\_pritisnemo\_na\_tipko\_[tipka]* in *predvajaj\_zvok\_[zvok]*.

ko pritisnemo na tipko de predvajaj zvok doge

# [5. korak]

V naslednjem koraku učenci uvozijo zvoke še od preostalih živali. Ko imajo zvoke shranjene na svojem računalniku, jih enostavno primejo, povlečejo in spustijo v zavihek *Zvoki*.



Ko so vsi zvoki uvoženi, jih učenci dodajo v kodo pod zavihkom *Skripte*. Pri tem uporabijo bloka *ko\_pritisnemo\_na\_tipko\_[tipka]* in





#### predvajaj\_zvok\_[zvok].

ko pritisnemo na tipko c	ko pritisnemo na tipko ov
predvajaj zvok cow -	predvajaj zvok pig -
ko pritisnemo na tipko h	ko pritisnemo na tipko s 🔹
predvajaj zvok horse	predvajaj zvok sheep *

#### [6. korak]

Sedaj pa učenci dodajo še posneman zvok. Učenci posnemajo pozdrav gospodarice: »Dobrodošli na moji kmetiji!« in ga dodajo na začetek igre. V zavihku *Zvoki* najdejo rdeči gumb. S klikom nanj se odpre snemalnik zvoka. Za snemanje kliknejo na prvi gumb s krogcem, s klikom na drugi gumb s kvadratkom vstavijo snemanje, za predvajanje posnemanega zvoka pa kliknejo na tretji gumb s trikotnikom. Ko so učenci zadovoljni s posnemanem zvokom, ga shranijo s klikom na gumb *Shrani*.

ନ୍ଦ୍ର Kr	netija_SLO
e 🦉	Woman farmer
Skript	e Obleke Zvoki
Zvok uvoziš (	tako, da da povlečaš sem acord a new sound
	Sound Recorder
	Shrani Prekliči

Ko je pozdrav gospodarice posneman, ga učenci dodajo v kodo gospodarice pod zavihkom *Skripte*. Pri tem uporabijo blok *predvajaj\_zvok\_[zvok]*.





	ko kliknemo na 🍋
	predvajaj zvok recording *
	čakaj 🚯 sekund.
	reci Čeželiš slišati psa pritisni tipko "D"i za (3) sekund.
	čakaj 1 sekund.
	Cakat (1) sekund
	reci Čeželiš slišati ovco, pritisni tipko "S"  za (3) sekund.
	čakaj 🌗 sekund.
	reci Čeželiš slišati pujsa, pritisni tipko "P" za (3) sekund.
	čakaj 🕦 sekund.
	reci Če želiš slišati konja, pritisni tipko Hilj za (3) sekund.
[Dodatne na	aloge]
Učenci lahko	o igri dodajo poljubne elemente, like (npr. kmeta, traktor,
petelina ipd.) in jim doda zvoke.	
Refleksija in	vrednotenje
Učenci ponc	ovijo in utrdijo pridobljeni znanje:
• kako	so dodali zvok v kodo:
<ul> <li>kake</li> <li>kater</li> </ul>	re bloke so uporabili pri dodajanju zvokov v kodo:
<ul> <li>kater</li> </ul>	re kontrolne bloke so uporabili;
[Končna koc	
Gospodarica	a kmetije
ka kliknem	no na 🔤
predvajaj a	zvok recording =
čakaj 3	sekund.
reci Če želiť	sišati psa pritisni tipko: "Dil za (3) sekund.
čakaj (1)	sekund
reci Če želiš	sisiati kravo pritisni tinko "C"I za S sekund.
čakaj (1)	sekund.
reci Če želiš	sišati ovco pritisni tipko "S", za (3) sekund.
čakaj (1	sekund.
reci Čerželik	sišati pujsa pritisni tipko "P" za 3 sekund.
čakaj (1)	sekund.
and the second se	

reci Če želiš slišati konja, pritisni tipko"H"! za 3 sekund.





	Ozadje
	ko pritisnemo na tipko o 💌 predvajaj zvok cow = predvajaj zvok pig =
	ko pritisnemo na tipko h = predvajaj zvok horse = predvajaj zvok sheep =
	ko pritisnemo na tipko d <del>-</del> predvajaj zvok dog -
Učni pripomočki,	<ul> <li>Primer celotne aktivnosti v Snap!-u:</li> <li>https://cpap.barkelov.edu/project2user=tadoja%</li> </ul>
sredstva za učitelja	project=Kmetija_SLO
	<ul> <li>Spletna stran s prostodostopnimi slikami:</li> </ul>
	<ul> <li><u>https://pixabay.com/</u></li> <li>Spletna stran s prostodostopnimi zvoki:</li> </ul>
	https://www.zapsplat.com/
	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>
Učni pripomočki za	<ul> <li>Predloga aktivnosti v Snap!-u:</li> </ul>
učenca	https://snap.berkeley.edu/project?user=tadeja&
	<ul> <li>Navodila za učenca (C4G5_NavodilaZaUcenca.docx)</li> </ul>





# Učni scenarij 6 – Kameleon na počitnicah

Naslov učnega	Kameleon na počitnicah
scenarija	
Pričakovano	predznanje ni potrebno
programersko	
predznanje	
Učni cilji	Splošni učni cilji:
	<ul> <li>premikanje preko dogodkov,</li> </ul>
	• zaznavanje ene barve,
	<ul> <li>Boolean vrednosti v logičnih izjavah,</li> </ul>
	<ul> <li>definiranje, razlikovanje, dinamično preverjanje in odzivanje na dve različni stanji igre,</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>učenec implementira premikanje objekta s smernimi tipkami in pri tem uporabi dogodke, pri tem pa upošteva omejitve,</li> </ul>
	<ul> <li>učenec uporabi blok za zaznavanje barve, da pridobi boolean vrednost, ki pove ali se objekt dotika neke barve,</li> </ul>
	<ul> <li>učenec ve, da lahko določi stanje objekta preko barv, ki se jih ta dotika,</li> </ul>
	<ul> <li>učenec razlikuje med dvema (enostavna različica) oz. petimi (zahtevnejša različica) različnimi stanji objekta in jih zna izraziti z logičnimi izjavami,</li> </ul>
	<ul> <li>učenec ve, da se pozicija objekta v igri dinamično spreminja in uporabi neskončno zanko, da z njo periodično preverja njegovo trenutno stanje,</li> </ul>
	<ul> <li>učenec uporabi pogojni blok "če-sicer", da glede na trenutno pozicijo objekta poda ustrezen odziv.</li> </ul>
Cilji, naloge in kratek	Kratek opis: Izdelajte preprosto igro, v kateri bo glavni objekt
opis aktivnosti	spreminjal svojo obleko glede na barvo ozadja na njegovi trenutni
	poziciji.
	Naloge: Učenci naj sprogramirajo kameleona, ki bo spreminjal svoj
	izgled (obleko) v dveh (enostavna različica): 1) ko bo plaval v morju,




	bo postal modre barve in rekel: "Kopam se v morju"; 2) ko bo na plaži			
	bo rjave barve in bo rekel: "Sončnim se na plaži"; oz. petih			
	(zahtevnejša različica) različnih situacijah: 1) ko bo plaval v morju, bo			
	postal modre barve in rekel: "Kopam se v morju"; 2) ko se bo nahajal			
	med morjem in plažo bo spremenil barvo v kombinacijo modre in			
	rjave in rekel: "Nahajam se med morjem in plažo."; 3) ko bo na plaži			
	bo rjave barve in bo rekel: "Sončnim se na plaži"; 4) ko se bo nahajal			
	med plažo in gozdom bo spremenil barvo v kombinacijo rjave in			
	zelene in rekel "Nahajam se med gozdom in plažo"; 5) v gozdu bo			
	spremenil svojo barvo v zeleno in rekel: "Hladim se v senci dreves".			
	Učenci bodo spoznali blok za zaznavanje barve in se naučili kako ga			
	lahko uporabijo v logičnih izrazih z namenom, da razlikujejo med			
	dinamično spreminjajočimi stanji igre, ki so odvisna od trenutne			
	pozicije glavnega lika (kameleona) in pri vsakem podali ustrezen			
	odziv.			
Trajanje aktivnosti	45 minut			
Trajanje aktivnosti Učne strategije in	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov			
Trajanje aktivnosti Učne strategije in metode	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov			
Trajanje aktivnosti Učne strategije in metode Učne oblike	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi			
Trajanje aktivnosti Učne strategije in metode Učne oblike	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje)			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) [Enostavna različica]			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) [Enostavna različica] Kameleon se je odpravil na poletne počitnice. Rad se kopa v morju in			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) [Enostavna različica] Kameleon se je odpravil na poletne počitnice. Rad se kopa v morju in sonči na plaži. Kot vsak kameleon, tudi on spreminja barvo tako, da se			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) <b>[Enostavna različica]</b> Kameleon se je odpravil na poletne počitnice. Rad se kopa v morju in sonči na plaži. Kot vsak kameleon, tudi on spreminja barvo tako, da se zlije z barvo okolice, nato pa še pove kje se trenutno nahaja.			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) <b>[Enostavna različica]</b> Kameleon se je odpravil na poletne počitnice. Rad se kopa v morju in sonči na plaži. Kot vsak kameleon, tudi on spreminja barvo tako, da se zlije z barvo okolice, nato pa še pove kje se trenutno nahaja.			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	<ul> <li>45 minut</li> <li>aktivno učenje, kolaborativno učenje, reševanje problemov</li> <li>frontalni način podajanja snovi</li> <li>individualno delo/delo v parih/skupinsko delo</li> <li>(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)</li> <li>[Enostavna različica]</li> <li>Kameleon se je odpravil na poletne počitnice. Rad se kopa v morju in sonči na plaži. Kot vsak kameleon, tudi on spreminja barvo tako, da se zlije z barvo okolice, nato pa še pove kje se trenutno nahaja.</li> <li>[1. korak]</li> </ul>			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	<ul> <li>45 minut</li> <li>aktivno učenje, kolaborativno učenje, reševanje problemov</li> <li>frontalni način podajanja snovi</li> <li>individualno delo/delo v parih/skupinsko delo</li> <li>(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)</li> <li>[Enostavna različica]</li> <li>Kameleon se je odpravil na poletne počitnice. Rad se kopa v morju in sonči na plaži. Kot vsak kameleon, tudi on spreminja barvo tako, da se zlije z barvo okolice, nato pa še pove kje se trenutno nahaja.</li> <li>[1. korak]</li> <li>Učencem damo navodilo, da spremenijo izgled ozadja tako, da je</li> </ul>			
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	<ul> <li>45 minut</li> <li>aktivno učenje, kolaborativno učenje, reševanje problemov</li> <li>frontalni način podajanja snovi</li> <li>individualno delo/delo v parih/skupinsko delo</li> <li>(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)</li> <li>[Enostavna različica]</li> <li>Kameleon se je odpravil na poletne počitnice. Rad se kopa v morju in sonči na plaži. Kot vsak kameleon, tudi on spreminja barvo tako, da se zlije z barvo okolice, nato pa še pove kje se trenutno nahaja.</li> <li>[1. korak]</li> <li>Učencem damo navodilo, da spremenijo izgled ozadja tako, da je razdeljen na dva dela iste barve. Prva polovica naj bo modra, ki bo</li> </ul>			





Lahko jim damo dodatno navodilo, da v sliko ozadja vključijo tudi druge elemente, da bi naredili ozadje bolj realistično. Vključijo lahko slike: valov, školjk, peščenih gradov, senčnikov, ipd. Opozorimo jih na to, da ne uporabijo predmetov, ki bi bili večji od glavnega lika in bi bili popolnoma druge barve od ozadja, saj v tem primeru blok za zaznavanje barve ne bo mogel ugotoviti, na katerem delu zaslona se nahaja kameleon.



### [2. korak]

Učenci naj narišejo ali poiščejo sliko kameleona na spletu (v primeru da sliko poiščejo, jih opozorimo, da mora imeti slika ustrezno licenco). Sliko kameleona naj pobarvajo modro (morje) oz. rjavo (plaža).



## [3. korak]

Najprej morajo sprogramirati premikanje kameleona v štirih smereh z uporabo tipk. Kombinacijo tipk lahko določijo sami (npr. smerne tipke, WASD, ipd.). V tem predlogu učnega scenarija privzemamo, da so se to naučili v eni od prejšnjih aktivnosti. Učence na tem mestu le opozorimo, da morajo upoštevati omejitve pri premikanju oz. da se kameleon ne sme premakniti izven zaslona in da to dosežemo z uporabo bloka "odbij se, če si na robu".

Gibanje lahko naredimo bolj realistično, če pred premikom kameleona obrnemo v smeri tega premika. To dosežemo z uporabo bloka "obrni





se v smeri", kjer med možnostmi izberemo ustrezno smer.



### [4. korak]

Učence seznanimo s konceptom zaznavanja barve oz. barv, ki se jih glavni lik dotika. Z blokom "se dotika <barva>" lahko dobimo podatek v obliki boolean vrednosti (res oz. ni res) o tem ali se objekt dotika izbrane barve. To, da dobimo podatek v obliki boolean vrednosti nam omogoča, da blok uporabimo v pogojnem bloku in s tem določimo ali oz. kdaj se bodo izvedli bloki v telesu pogojnega bloka.

Z učenci se pogovorimo o tem katere so različne možnosti pozicije kameleona glede na barvo ozadja na sceni in kako jih lahko izrazimo preko bloka za zaznavanje barve.

Možnosti sta dve:

- 1. kameleon se dotika modre barve -> se dotika <modra>?
- 2. kameleon se dotika rjave barve -> se dotika <rjava>?

Ko se dotika določene barve moramo spremeniti njegov izgled (obleka) in uporabiti blok "reci" iz skupine ukazov "Izgled", da bo povedal kje se trenutno nahaja. Izgled objekta spremenimo tako, da preklapljamo med različnimi oblekami objekta. To naredimo tako, da uporabimo blok "Preklopi na obleko", ki ga najdemo v skupini ukazov "Izgled" in iz spustnega menija v bloku izberemo želeno obleko.

Kameleon se nahaja bodisi na modri podlagi, bodisi na rjavi podlagi, zato lahko za ločevanje uporabimo blok "če-sicer", saj če se ne dotika npr. rjave barve, posledično vemo, da se prav gotovo dotika modre, saj sta barvi samo dve:







### [5. korak]

Učencem povemo, da v situacijah, ko moramo nekatere ukaze izvrševati ves čas izvajanja programa, uporabimo neskončno zanko. Vse kar bomo napisali pod neskončno zanko se bo periodično izvrševalo dokler se ne bo program zaključil. Preko diskusije z učenci razložimo, da je ta situacija točno taka, da jo moramo realizirati z neskončno zanko. Pozicija kameleona se namreč ves čas spreminja, prav tako pa ne moremo predvideti kako ga bo igralec premikal, zato moramo to kje se trenutno nahaja, ves čas preverjati.

## [Končna koda]



## [Zahtevnejša različica]

Kameleon se je odpravil na poletne počitnice. Kot vsak kameleon, tudi on spreminja barvo tako, da se zlije z barvo okolice, nato pa še pove





kje se trenutno nahaja. Rad se kopa v morju, sonči na plaži in ko postane prevroče se gre ohladit v senco dreves v gozdu.

### [1. korak]

Učencem naročimo naj spremenijo izgled ozadja tako, da bo razdeljen na tri dele iste barve. Vsak del bo predstavljal različen del scene: modra bo predstavljala morje, rjava predstavlja plažo, zelena pa gozd. Učenci lahko po svoji izbiri na ozadje dodajo še dodatne predmete, kot npr.: valove, školjke, peščene gradove, senčnike, drevesa, ipd. da jo naredijo bolj realistično. Pri tem morajo biti pozorni, da ne uporabijo predmetov, ki bi bili večji od glavnega lika in bi bili popolnoma druge barve od ozadja, saj v tem primeru blok za zaznavanje barve ne bo mogel ugotoviti, na katerem delu zaslona se nahaja kameleon.



## [2. korak]

Učenci naj narišejo ali poiščejo sliko kameleona na spletu (v primeru da sliko poiščejo, jih opozorimo, da mora imeti slika ustrezno licenco). Sliko kameleona naj pobarvajo modro (morje), rjavo (plaža), zeleno (gozd), modro-rjavo (območje med morjem in plažo) in rjavo-zeleno (območje med plažo in gozdom), da bodo s tem ponazorili vse možnosti v katerih se lahko znajde kameleon.







[3. korak]

Najprej morajo sprogramirati premikanje kameleona v štirih smereh z uporabo tipk. Kombinacijo tipk lahko določijo sami (npr. smerne tipke, WASD, ipd.). V tem predlogu učnega scenarija privzemamo, da so se to naučili v eni od prejšnjih aktivnosti. Učence na tem mestu le opozorimo, da morajo upoštevati omejitve pri premikanju oz. da se kameleon ne sme premakniti izven zaslona in da to dosežemo z uporabo bloka "odbij se, če si na robu".

Gibanje lahko naredimo bolj realistično, če pred premikom kameleona obrnemo v smeri tega premika. To dosežemo z uporabo bloka "obrni se v smeri", kjer med možnostmi izberemo ustrezno smer.



[4. korak]

Učence seznanimo s konceptom zaznavanja barve oz. barv, ki se jih glavni lik dotika. Z blokom "se dotika <barva>" lahko dobimo podatek v obliki boolean vrednosti (res oz. ni res) o tem ali se objekt dotika izbrane barve. To, da dobimo podatek v obliki boolean vrednosti nam omogoča, da blok uporabimo v pogojnem bloku in s tem določimo ali oz. kdaj se bodo izvedli bloki v telesu pogojnega bloka.

Z učenci se pogovorimo o tem katere so različne možnosti pozicije kameleona glede na barvo oz. barve ozadja na sceni in kako jih lahko izrazimo preko bloka za zaznavanje barve.

Ugotovimo, da imamo pet možnosti:

- Kameleon se v celoti dotika modrega dela ozadja -> se dotika <modra>?
- 2. Kameleon se v celoti dotika rjavega dela ozadja -> se dotika





<rjava>? 3. Kameleon se v celoti dotika zelenega dela ozadja -> se dotika <zelena>? 4. Kameleon je delno na modrem, delno na rjavem delu ozadja -> se dotika <modra>? IN se dotika <rjava>? 5. Kameleon je delno na rjavem, delno na zelenem delu ozadja -> se dotika <rjava>? IN se dotika <zelena>? Ko se dotika določene barve moramo spremeniti njegov izgled (obleka) in uporabiti blok "reci" iz skupine ukazov "Izgled", da bo povedal kje se trenutno nahaja. Izgled objekta spremenimo tako, da preklapljamo med različnimi oblekami objekta. To naredimo tako, da uporabimo blok "Preklopi na obleko", ki ga najdemo v skupini ukazov "Izgled" in iz spustnega menija v bloku izberemo želeno obleko. Najprej to naredimo za enostavnejše situacije, ko je kameleon v celoti na delu ozadja iste barve. Nato oblikujemo logično izjavo z uporabo operatorja IN, saj želimo preveriti ali se dotika dveh barv hkrati, torej ene IN druge. se dotika se dotika Semimedimoriemiintobalo ci Sem med obalo in oozdom Če združimo zgornja bloka in jih postavimo pod dogodkovni blok za

začetek programa ("ko kliknemo na zeleno zastavico") ugotovimo, da se bosta pogojna bloka izvedla natanko enkrat. Učencem pojasnimo, da se bo pozicija kameleona zaradi poljubnega premikanja med igro spreminjala, zato ni dovolj, da le na začetku preverimo oba pogoja,





ampak moramo to početi ves čas izvajanja programa.

[5. korak]

Učencem povemo, da v situacijah, ko moramo nekatere ukaze izvrševati ves čas izvajanja programa, uporabimo neskončno zanko. Vse kar bomo napisali pod neskončno zanko se bo periodično izvrševalo dokler se ne bo program zaključil. Preko diskusije z učenci razložimo, da je ta situacija točno taka, da jo moramo realizirati z neskončno zanko. Učence prav posebej upozorimo na vrstni red preverjanja pogojev. Če bi postavili pogojni blok, ki preverja, če se kameleon nahaja na dveh barvah hkrati, pred blok, ki ugotavlja, če se dotika ene barve, bi se upošteval pogoj, ki je napisan kasneje.

[Končna koda]







## [Učenci kodo prilagodijo]

Če želimo poenostaviti aktivnost, lahko pripravimo nekatere dele kode vnaprej v predlogi, ki jo učenci nato dopolnijo.

Učenci, ki so sledili priporočenemu sosledju učnih scenarijev, so se že naučili kako premikati objekt s tipkami. Zato predlagamo, da to kodo vključite v predlogo. Lahko pa kodo premikanja tudi prilagodijo po svoje, npr. namesto smernih tipk, za premikanje uporabijo kombinacijo WASD.



En od ključnih učnih ciljev v aktivnosti je razumevanje pomena neskončne zanke in prepoznavanje situacij, ki jih moramo reševati z njeno uporabo. V predlogo lahko vključimo del kode, ki se bo izvajala periodično, ostalo pa dopolnijo sami. Na ta način bodo še vedno imeli priložnost razumeti koncept neskončne zanke, čeprav ne bodo definirali vseh situacij sami, ampak le dopolnili manjkajoče. Koda v





	predlogi paj vključuje dve pomensko različni situaciji: 1) objekt je v		
	prediogi naj vkijucuje dve pomensko različni sltuaciji. 1) objekt je v		
	celoti na eni barvi, 2) objekt se hkrati dotika dveh različnih barv.		
	Predlagan del kode v predlogi:		
	ko kliknemo na za vedno če se dotika ? reci Sončimsena:plaži. Preklopi na obleko kameleon_rjav če se dotika ? in se dotika ? reci Semimedimorjemimobalo. Preklopi na obleko kameleon_oranzen_moder		
Orodja in viri za	<ul> <li>Enostavna različica v programu Snap!:</li> </ul>		
učitelie	https://snap.berkeley.edu/project?user=zapusek&project=ka		
ucitelje	<u>meleon enostavni SLO</u>		
	<ul> <li>Zahtevnejša različica v programu Snap!:</li> </ul>		
	<u>https://snap.berkeley.edu/project?user=zapusek&amp;project=ka</u> meleon SLO		
	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> </ul>		
	<ul> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>		
Viri/gradiva za	Predloga le s kalemeonom in ozadjem:		
učence	https://snap.berkeley.edu/project?user=zapusek&project=ka		
	meleon SLO predloga		
	<ul> <li>Predloga s slikovnim materialom za enostavno različico: <u>https://snap.berkeley.edu/project?user=zapusek&amp;project=ka</u> <u>meleon enostavni slike SLO</u></li> </ul>		
	<ul> <li>Predloga s slikovnim materialom za zahtevnejšo različico: <u>https://snap.berkeley.edu/project?user=zapusek&amp;project=ka</u> <u>meleon zahtevnejsi predloga slike SLO</u></li> </ul>		
	<ul> <li>Delno rešena predloga za učence: <u>https://snap.berkeley.edu/project?user=zapusek&amp;project=ka</u> <u>meleon_predloga_SLO</u></li> </ul>		

Učni scenarij 7 – Pomagaj princu in princeski najti svoje živali





Naslov učnega	Pomagaj princu in princeski najti svoje živali		
scenarija			
Pričakovano	Dodajanje besedila za lik		
programersko	Premikanje lika s tipkami in uporabo dogodkov		
predznanje	Uporaba pogoja <i>če se lik dotika</i>		
	Uporaba dogodkov		
Učni cilji	<ul> <li>Splošni učni cilji:</li> <li>Pogoj če se lik dotika določene barve</li> <li>Premikanje po koordinatah</li> <li>Svinčnik dvignjen, spuščen</li> <li>Barva svinčnika</li> </ul> Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje: <ul> <li>Učenec uporabi pogojni stavek za preverjanje, ali se lik dotika</li> </ul>		
	<ul> <li>določene barve in mu v primeru izpolnjenega pogoja ne dovoli nadaljevanja poti</li> <li>Učenec nastavi začetne x in y koordinate lika</li> <li>Učenec uporabi dvignjen / spuščen svinčnik za risanje črte / poti</li> <li>Učenec spremeni barvo svinčnika glede na par, ki ga povezuje</li> <li>Učenec ve, da mora na začetku izbrisati vse poti, ki so ostale od prej</li> </ul>		
Cilji, naloge in	Kratek opis: Dekle mora pomagati princeski najti svojo mačko in princu najti		
kratek opis	svojega psa. To stori tako, da gre do princeske in ji z risanjem črte pokaže pot		
aktivnosti	do mačke, na enak način pokaže tudi princu pot do psa. Na svoji poti se mora		
	deklica izogniti srečanju med živalmi, zato se poti ne smejo križati.		
	Naloge: V prvem koraku morajo učenci izbrati ustrezno ozadje (labirint), nato		
	dodajo 5 likov: dekle, princesko, princa, mačko in psa. Nato napišejo kodo za		
	premikanje s tipkami (z uporabo dogodkov), kjer pa morajo dekletu		
	preprečiti, da stopi na travo. Sledi risanje s svinčnikom in spreminjanje barve		
	z uporabo dogodkov. Prav tako morajo napisati kodo, ki na začetku izbriše		
	vse poti, in napisati začetna navodila, ki jih poda dekle.		
	Cilj: Učenci bodo spoznali risanje s premikanjem lika. Poleg tega se bodo		
	naučili, kako z uporabo pogojnega stavka preprečiti liku prosto gibanje po		
	celotnem labirintu.		
Trajanje	30 min		
aktivnosti			





Učne strategije	Aktivno učenje, učenje z izdelavo iger, reševanje problemov		
in metode			
Učne oblike	Frontalna učna oblika		
	Individualna učna oblika		
Povzetek	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)		
učnega			
procesa	Za lažji začetek lahko damo učencem predlogo, v kateri imajo na voljo:		
	<ul> <li>Ozadje labirinta</li> <li>Lik dekleta</li> <li>Kodo za premikanje v eno smer</li> </ul>		
	Dekle se odloči, da bo pomagalo princesi najti svojo mačko in princu svojega		
	psa. To bo storila tako, da jima bo pokazala (narisala) pot do njinuh živali. V		
	izogib srečanju živali mora poskrbeti, da se poti ne bodo križale, označila pa		
	bo tudi vsako pod z drugo barvo.		
	[Korak 1]		
	Za realizacijo pogoja »če se dotika barve« mora biti ozadje (trava) enobarvna		
	ali pa mora imeti pot enobarvni okvir, kot v našem primeru. Da se izognemo		
	tem »težavam« pri iskanju ustreznega ozadja, ponudimo učencem zgornje		
	ozadje.		
	[Korak 2]		
	Učenci imajo dan lik dekleta, najti morajo še ostale 4 in jih postaviti v labirint.		







Kodo dodamo k prejšnji. To je sedaj koda za premik v desno:

















	<ul> <li>Vsak naj riše pot z drugo barvo.</li> <li>Prilagodi začetna navodila.</li> <li>Dodaj navodila za premikanje lika in risanje s klikom na lik.</li> </ul>
Učni	Primer celotne aktivnosti v Snap!-u:
pripomočki,	<u>https://snap.berkeley.edu/project?user=mateja&amp;project=C4G7</u> Pomagaj princu in princeski%20-%20Cela
sredstva za	<ul> <li>Primer dodatne aktivnosti v Snap!-u:</li> </ul>
učitelja	https://snap.berkeley.edu/project?user=mateja&project=C4G7
	<ul> <li>Pomagaj princu in princeski%20-%20Cela%20%2B%20Dodatek</li> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>
Učni pripomočki za učenca	<ul> <li>Predloga aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=mateja&amp;project=C4G7</u> <u>Pomagaj princu in princeski</u></li> <li>Navodila za učenca (C4G7_NavodilaZaUcenca.docx)</li> </ul>





# Učni scenarij 8 – Risanje s kredo

Naslov	Risanje s kredo
učnega	
scenarija	
Pričakovano	Dodajanje besedila za lik
programersk	Risanje s svinčnikom (svinčnih spuščen, dvignjen, barva)
o predznanje	Premikanje lika
	Uporaba zank
	Uporaba dogodkov
Učni cilji	<ul> <li>Splošni učni cilji:</li> <li>Zanka ponovi</li> <li>Obrat za 90 stopinj</li> <li>Obrni se v smeri</li> <li>Menjava ozadja</li> </ul>
	<ul> <li>Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:</li> <li>Učenec uporabi zanko ponovi, ko se ista koda ponovi 2/4 krat</li> <li>Učenec uporabi <i>obrat za 90 stopinj</i> pri risanju kvadrata, pravokotnika, črke »T«</li> <li>Učenec razume ukaz <i>obrni se v smeri 90</i></li> <li>Učenec zna zamenjati ozadje v kombinaciji z uporabo dogodka <i>ko pritisnemo na tipko</i></li> </ul>
Cilji, naloge	Kratek opis: Igralec mora s pritiskom na tipko povezati tičke v tri različne oblike –
in kratek	kvadrat, pravokotnik in črko »T«.
opis	Naloge: Učenci izberejo ozadje <i>tablaKvadrat</i> in napišejo kodo za risanje
aktivnosti	kvadrata. Začetna pozicija je točka A. Pri risanju kvadrata ponovijo določene
	korake 4 krat, zato namesto pisanja iste kode 4 krat, uporabijo zanko ponovi 4
	krat. Nato napišejo kodo za risanje pravokotnika, kjer uporabijo zanko ponovi 2
	krat. V zadnjem koraku povečejo točke v črko T. Ko je mogoče, uporabijo zanko.
	Zamenjati morajo tudi ozadje za pravokotnik in črko T.
	Cilji: Učenci se bodo naučili risanja različnih oblik s kodo. Naučili se bodo
	uporabo zanke ponovi za krajšanje kofe in menjavo ozadja.
Trajanje	60 min
aktivnosti	





Učne	Aktivno učenje, učenje z izdelavo iger, reševanje problemov		
strategije in			
metode			
Učne oblike	Frontalna učna oblika		
	Individualna učna oblika / Delo v paru		
Povzetek	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)		
učnega			
procesa	Učenci imajo podana:		
	<ul><li>3 ozadja s točkami, ki jih morajo povezati</li><li>Lik krede</li></ul>		
	Kreda bi rada narisala kvadrat, pravokotnik in lik oblike črke T, ampak se ne zna		
	premikati in obračati. Pomagaj kredi in ji povej, kaj mora narediti!		
	[Korak 1]		
	NARISI KVADRAT		
	DC		
	e A		
	Učenci začnejo z zgornjim ozadjem. Napisali bodo kodo za premikanje krede.		
	Kreda začne v točko A, se premake X korakov do točke B, se obrne za 90 stopinj v		
	levo, se premakne za X korakov do točke C, se obrne za 90 stopinj v levo, se		
	premakne X korakov do točke D, se obrne za 90 stopinj v levo in se premakne za		
	X korakov to točke A (in se obrne za 90 stopinj v levo).		





premakni se (150) korakov
čakaj 🕕 sekund.
obrni se 👌 🧐 stopinj
čakaj 🕕 sekund.
premakni se (150) korakov
čakaj 🕕 sekund.
obrni se 👆 🤫 stopinj
čakaj 🕦 sekund.
premakni se (150) korakov
čakaj 🕕 sekund.
obrni se 👌 🤫 stopinj
čakaj 🕕 sekund.
premakni se (150) korakov
čakaj 🕕 sekund.
obrni se 👌 🤫 stopinj
čakaj 🕕 sekund.

Uporabaili smo ukaz *obrni se za 90 stopinj*. Druga možnost je obračanje z ukazi *obrni se v smeri 0, 90, 180 in -90*, ampak je možnost 1 lažja, saj je ukaz vedno isti (obrni se za 90 stopinj), prav tako pa lahko uporabimo tudi zanko za ponovitev 4ih enakih korakov, česar v možnosti 2 ne moremo narediti.

Ukaz *čakaj 1 sekund* je dodan, da se pri risanju vidijo vsi koraki (animacija). Brez tega ukaza se cela koda izvede v trenutku in koraki niso vidni. Učenci naj poskušajo najprej brez uporabe ukaza *počakaj*, da vidijo njegov pomen. Učenci naj sami pomislijo, kako bi skrajšali kodo, če je to mogoče. Se kateri deli ponavljajo? Odgovor je da. Namesto, da pišemo isto kodo 4 krat, pri programiranju uporabimo zanko *ponovi*.



Če želimo, da kreda riše pot, moramo pred zanko dodati še ukaz *svinčnik spuščen*.

svinčnik spuščen

Če želimo, da se lik krede ne vrti med obračanjem, kliknemo na ukaz ne vrti.







### [Korak 2]

Za začetek izvedbe kode lahko uporabimo ukaz *ko pritisnemo na tipko K*. Prav tako lahko nastavimo barvo krede, spustimo svičnik (v primeru, da je ostal v prejšnji igri spuščen), zbrišemo (izbriše risanje iz prejšnje igre), nastavimo začetne koordinate kredi (v točki A).

Včasih se zgodi, da program prekinemo med izvajanjem in ostane kreda obrnjena v napačno smer. Težava nato nastane, ko program zaženemo, in se kreda namesto v desno, premakne v levo, gor ali dol. Da se temu izognemo, na začetku nastavimo ukaz *obrni se v smeri 90*.



## [Korak 3]

Sledi koda za pravokotnik. Najprej moramo zamenjati ozadje. To naredimo v dveh korakih:

a) Kliknemo na ozadje (imenovano tabla)



















S klikom na tipko P se ozadje spremeni v zgornje ozadje. Podobno kot prej, napišemo kodo za premikanje krede, ki bo povezala točke in narisala pravokotnik. Učenci lahko prejšnjo kodo podvojijo in ustrezno spremenijo. Spremeniti morajo zanko ponovi, ki se bo sedaj izvedla 2 krat.



## [Korak 5]

Po pravokotniku sledi še zadnja naloga in sicer bodo učenci povezali točke v obliki črke T. Spremeniti morajo torej ozadje, kar pomeni, da ponovijo [Korak 3] in spremenijo le črko P v T in ozadje *tablaPravokotnik* v *tablaT*.

a) Ko pritisnemo na tipko T se ozadje Preklopi na obleko tablaT.









S pritiskom na tipko T se ozadje spremeni v zgornje. Podobno kot prej moramo točke povezati v črko T. Učenci lahko kopirajo prejšnjo kodo in jo ustrezno spremenijo.

Spremeniti morajo začetne koordinate, ki niso enake prejšnjim. Kako določijo pravilne začetne koordinate že vedo.

Nato točke povežejo v črko T. Ugotoviti morajo ustrezno število korakov (v pomoč imajo navodila za učenca). Možna rešitev:

ko pritisnemo na tipko 1 💌 nastavi barvo svinčnika na zbriši svinčnik dvignjen obrni se v smeri 🧐 🗸 pojdi na x: -56) y: -138 svinčnik spuščen premakni se 60 korakov akaj (1) sekund. obrni se 👌 🧿 stopinj kaj 🚺 sekund. premakni se 185) korakov kaj 1) sekund. obrni se 👌 🧕 stopinj čakaj 🚺 sekund. onovi (2) krat premakni se 60 korakov čakaj 1 sekund. obrni se 🌜 🧐 stopinj čakaj 1 sekund. premakni se 180 korakov čakaj 🚺 sekund. obrni se 👌 90 stopinj čakaj 🚺 sekund. premakni se 60 korakov iakaj (1) sekund. obrni se 👌 🤫 stopinj akaj 🚺 sekund. premakni se 👩 korakov ikaj 🚺 sekund. obrni se 👌 🧐 stopinj akaj 🚺 sekund. premakni se (185) korakov

[Korak 7] Na koncu dodamo še kodo za spremembo ozadja na začetno ozadje za risanje











	<ul> <li>Dodaj novo ozadje in nariši točke / oglišča.</li> <li>Napiši kodo, ki poveže oglišča. Ozadje lahko narišeš sam ali pa uporabiš dano ozadje.</li> </ul>
Učni	<ul> <li>Primer celotne aktivnosti v Snap!-u:</li> </ul>
pripomočki,	https://snap.berkeley.edu/project?user=mateja&project=C4G8
sredstva za	<u>Risanje s kredo</u>
učitelja	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana:</li> </ul>
	MK.
Učni	Predloga aktivnosti v Snap!-u:
pripomočki	https://snap.berkeley.edu/project?user=mateja&project=C4G8
za učenca	<u>Risanje s kredo%20-%20Delno</u>
	<ul> <li>Navodila za učenca (C4G8_NavodilaZaUcenca.docx)</li> </ul>





# Učni scenarij 9 – Pobiranje smeti in čiščenje parka

Naslov učnega	Pobiranje smeti in čiščenje parka		
scenarija			
Pričakovano	Določanje začetnih koordinat		
programersko	Nastavljanje velikosti lika		
predznanje	Dodajanje besedila za lik		
	Premikanje lika s tipkami in uporabo dogodkov		
	Uporaba pogoja <i>če se lik dotika</i>		
Učni cilji	Splošni učni cilji: • Spremenljivke • Prikaži in skrij lik • Podvoji lik • Podvoji del kode • Pogojni stavek		
	<ul> <li>Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:</li> <li>Učenec uporabi spremenljivko za štetje pobranih smeti</li> <li>Učenec zna skriti lik, ko se ga dekle dotakne, in prikazati lik na začetku igre</li> <li>Učenec zna podvojiti lik (iz 1 steklenice naredi npr. 4)</li> <li>Učenec zna podvojiti del kode (npr. kodo steklenice podvoji in prenese na kodo papirja)</li> <li>Učenec zna uporabiti pogojni stavek za preverjanje, ali je lik prikazan in ali so vse smeti pobrane</li> </ul>		
Cilji, naloge in	Kratek opis: Park je poln smeti in dekle se odloči, da ga bo očistilo. Ko		
kratek opis	pobere vse smeti, jih vrže v koš.		
aktivnosti	<ul> <li>Naloge: Učenci nastavijo dekletu začetne koordinate. Igra se konča, ko dekle pobere vse smeti in jih odvrže v koš. Za preverjanje, ali so vse smeti v košu, bodo učenci uporabili spremenljivko (1 pobrane smeti = 1 točka). Ko se dekle dotakne steklenice, se steklenica skrije in točke povečajo za 1. Ko pobere vse, gre do koša, koš pa ji sporoči, ali je pobrala vse smeti ali ne.</li> <li>Cilji: Učenci bodo spoznali rabo spremenljivk in podvajanje kode ali celotnega lika</li> </ul>		





Trajanje aktivnosti	45 min		
Učne strategije in	Aktivno učenje, učenje z izdelavo iger, reševanje problemov		
metode			
Učne oblike	Frontalna učna oblika		
	Individualna učna oblika		
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)		
procesa			
	Učencem je dano:		
	<ul> <li>Ozadje</li> <li>Lik dekleta (s kodo premikanja), lik steklenice, lik papirja in lik koša</li> </ul>		
	Dekle na sprehodu po parku opazi, da je v parku veliko smeti in se		
	odloči, da ga bo očistila. Ko pobere vse smeti, jih vrže v koš in nato		
	lahko uživa v lepem sončnem vremenu v čistem parku.		





[Korak 1]	
Poleg ozadja je poda	na tudi koda za premikanje s tipkami in pogojem za
dotikanje rjave barve	2:
ko pritisnemo na tipko puš obrni se v smeri -80 premakni se 10 korakov če se dotika ? premakni se -10 korako	ko pritisnemo na tipko puščica gor obrni se v smeri () premakni se () korakov ce se dotika ? premakni se () korakov ko pritisnemo na tipko puščica desno * obrni se v smeri () premakni se () korakov če se dotika ? premakni se () korakov ko pritisnemo na tipko puščica dol* obrni se v smeri () premakni se () korakov če se dotika ? premakni se () korakov
Na začetku nastavim	o začetne x in y koordinate dekleta. Učenci lahko
poljubno nastavijo x	in y, pomembno je le, da so znotraj poti. Dodajo
tudi začetna navodila	a, npr:
ka kilik pojdi r reci Pa	nemo na 19 na x: (-189) y: (-156) oberivse smeti in jihrodioži v koš zarsmeti, za (4) sekund.
[Korak 2]	
Za štetje pobranih sr	neti bomo uporabili spremenljivko.
Kaj je spremenljivka	
Spremenljivko si lahl	ko predstavljamo kot škatlo, v kateri shranjujemo
podatke. Lahko si pro	edstavljamo, da dekle pobere smeti in jih shrani v
spremenljivki, poime	novani <i>tocke</i> . Spremenljivka <i>tocke</i> torej šteje,
koliko smeti je dekle	pobralo.
Kako ustvarimo spre	menljivko?











angl. *shown*). Če je steklenica na tleh, jo poberemo »in damo v škatlo (spremenljivko)«. Pred tem je bila vrednost spremenljivke *tocke* 0, sedaj je 1. Ko poberemo steklenico, torej spremenimo vrednost spremenljivke (tocke) za 1. Ko steklenico poberemo, jo skrijemo.



Igro zaženemo s klikom na zeleno zastavico in preverimo, ali koda deluje. Z dekletom se pomaknemo do steklenice, ki mora izginiti in spremenljivka se mora povečati za 1. Nato želimo igrati še enkrat in ponovno kliknemo na zeleno zastavico. Kaj se zgodi? Kje je steklenica? Steklenica je skrita, saj smo jo v prejšnji igri pobrali (skrili). Zato moramo na začetku igre napisati kodo za prikaz steklenice:



[Korak 4]

Sedaj želimo imeti več steklenic. Lik steklenice lahko enostavno podvojimo z desnim klikom na lik in izberemo podvoji.



Podvojen lik se pojavi na ekranu. Kliknemo z miško nanj in ga povlečemo za željeno mesto v labirintu.

Postopek lahko ponovimo in podvojimo steklenico poljubno krat.





### [Korak 5]

Podobno kodo kot za steklenico želimo imeti tudi za lik papirja. Kodo preprosto primemo, povlečemo na željen lik in spustimo.



Na enak način prenesemo tudi kodo *ko kliknemo na zeleno zastavico – skrij*.

Sedaj lahko ponovimo [Korak 4] in podvajamo lik papirja, da imamo tudi več likov papirja po labirintu.

## [Korak 6]

Zadnji korak je še koda za lik koša za smeti. Lik je podan in poljubno ga lahko prestavijo na drugo mesto v labirintu.

Tudi ta koda se bo začela izvajati, ko se dekle dotakne koša.

Koš bo moral preveriti, ali so pobrane vse smeti ali ne. S pomočjo spremenljivke *tocke* bo to zelo enostavno. Recimo, da imamo v igri 8 likov smeti, zato bo koš preveril, ali je vrednost spremenljivke enaka 8. Če je, to pomeni, da so pobrane vse smeti, drugače pa ne. To bomo preverili z pogojnim stavkom *če* in dodali odziv, ali je igralec pobral vse smeti ali ne.

Ko je	se dotika Dekle - ?
ČG 🚺	tocke = 8
reci	Čestitkel·Pobrala·si-vse·smetil za (2) sekund.
sicer	
reci	Vrni-se, ko-pobereš-vse-smeti, za (2) sekund.





Dekle
ke kilknemo na pojdi na x: 189 y: 155 red: Poberivse smetin inrodioživ koš za smeti, za () sekund. natavi tocko na 0 ko pritisnemo na tipko pušče por obrni se v smeri 0 premakni se 10 korakov ke se dotika 2 premakni se 10 korakov ko pritisnemo na tipko pušče love obrni se v smeri 90 premakni se 10 korakov ke se dotika 2 premakni se 10 korakov
Steklenica / Papir
ko kilknemo na prikaži Ko je se dotika Dekle ? če shown? spremeni spremenijivko tocke za 1
Коš
Ko je se dotika Dekle ? če tocke = 8 reci Čestitkel Pobralarsi vsersmetil za 2 sekund. sicer reci Vrni se, korpobereši vsersmetil za 2 sekund.
[Dodatne naloge]





	Učenec doda dodatne naloge po svojih željah ali sledi spodnjim
	navodilom:
	<ul> <li>Dodaj še tretjo vrsto odpadkov (npr. bio).</li> <li>Koš naj izpiše: »Pobral/a si X steklenih, Y papirnatih in Z bio odpadkov«.</li> <li>Če igralec pobere vse smeti, koš reče »Čestitam, pobral/a si vse smeti!«.</li> <li>Če igralec ni pobral vseh smeti, mu koš pove, katerih smeti ni pobral vseh. Npr. »Nisi pobral vseha papirja«, »Nisi pobral vseh bio odpadkov«. Nato reče še »Vrni se, ko pobereš vse smeti«.</li> </ul>
Učni pripomočki,	Primer celotne aktivnosti v Snap!-u:
sredstva za učitelja	https://snap.berkeley.edu/project?user=mateja&project=C4G9
	<u>PobiranjeSmeti</u>
	<ul> <li>Primer dodatne aktivnosti v Snap!-u: https://snap.berkelev.edu/project?user_mateia&amp;project=C469</li> </ul>
	PobiranjeSmeti%20%2B%20Dodatek
	• Lajovic, S. (2011). Scratch. Nauči se programirati in postani
	računalniški maček. Ljubljana: Pasadena.
	Ljubljana: MK.
Učni pripomočki za	Predloga aktivnosti v Snap!-u:
učenca	https://snap.berkeley.edu/project?user=mateja&project=C4G9
	PobiranjeSmeti%20-%20Delno
	<ul> <li>Navodila za učenca (C4G9_NavodilaZaUcenca.docx)</li> </ul>





# Učni scenarij 10 – Nahrani mačke

Naslov učnega	Nahrani mačke
scenarija	
Pričakovano	<ul> <li>pogojni stavki (if, if-else blok)</li> </ul>
programersko	<ul> <li>izpis besedila (blok "reci")</li> </ul>
predznanje	
Učni cilji	Splošni učni cilji:
	<ul> <li>nastavljanje in povečevanje vrednosti spremenljivke,</li> </ul>
	<ul> <li>nastavljanje vrednosti spremenljivki znotraj/zunaj zanke,</li> </ul>
	<ul> <li>for zanka (oz. zanka ponovi n-krat),</li> </ul>
	<ul> <li>naključna števila,</li> </ul>
	<ul> <li>združevanje nizov,</li> </ul>
	<ul> <li>operatorji: logični, aritmetični</li> </ul>
	<ul> <li>uporabnikov vnos.</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>učenec prepozna situacijo, v kateri je smiselno uporabiti zanko, ki se ponovi natanko n-krat,</li> </ul>
	<ul> <li>učenec loči med nastavljanjem vrednosti spremenljivki pred zanko in med vsako ponovitvijo znotraj zanke,</li> </ul>
	<ul> <li>učenec zna uporabiti blok "vprašaj" in z njim dobi številko, ki jo je vpisal uporabnik,</li> </ul>
	<ul> <li>učenec zna uporabiti aritmetične operatorje in z njimi izračunati vrednost pravilnega odgovora,</li> </ul>
	<ul> <li>učenec zna uporabiti pogojni blok "če-sicer" in z njim preveri pravilnost odgovora, ki ga je podal uporabnik,</li> </ul>
	<ul> <li>učenec zna uporabiti spremenljivko za štetje pravilnih odgovorov.</li> </ul>
Cilji, naloge in kratek	Kratek opis: Izdelajte igro v kateri bo moral igralec opraviti deset
opis aktivnosti	računov množenja in šteti pravilne odgovore.
	Naloga: Izdelajte igro, v kateri bo oskrbnica mačjega zavetišča Marta





	ponavljajoče spraševala igralca o številu mačk, ki jih lahko nagrani v
	eni od desetih sob v zavetišču. Odgovor je odvisen od števila in
	velikosti posod v vsaki od sob. Število posod in njihova velikost se
	morata za vsako sobo posebej določiti naključno. Poleg tega želimo
	imeti števec, ki bo štel pravilne odgovore. Igra se naj začne tako, da lik
	oskrbnice razloži nalogo v igri, nato pa se igra začne. Igra se zaključi,
	ko oskrbnica desetkrat vpraša po številu nahranjenih mačk. Vsakič ko
	vpraša in uporabnik odgovori mu mora podati povratno informacijo o
	tem ali je pravilno odgovoril. Ko s spraševanjem konča mora igralcu
	podati povzetek njegove uspešnosti, tako da mu pove kolikokrat je
	odgovoril pravilno in kolikokrat napačno.
	Učenci se bodo preko aktivnosti seznanili s konceptom
	ponavljajočega prirejanja naključne vrednosti spremenljivki znotraj
	zanke in se naučili kako je ta situacija drugačna od prirejanja
	vrodnosti spromonlijuki izvon zanko. Drav tako so bodo povčili kako
	lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo
	lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne.
Trajanje aktivnosti	lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne. 45 minut
Trajanje aktivnosti Učne strategije in	<ul> <li>lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo</li> <li>pravilne.</li> <li>45 minut</li> <li>aktivno učenje, kolaborativno učenje, reševanje problemov</li> </ul>
Trajanje aktivnosti Učne strategije in metode	<ul> <li>lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne.</li> <li>45 minut</li> <li>aktivno učenje, kolaborativno učenje, reševanje problemov</li> </ul>
Trajanje aktivnosti Učne strategije in metode Učne oblike	Vrednosti spremenjivki izven zanke. Prav tako se bodo nadčin kako         lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo         pravilne.         45 minut         aktivno učenje, kolaborativno učenje, reševanje problemov         frontalni način podajanja snovi
Trajanje aktivnosti Učne strategije in metode Učne oblike	Iahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne. 45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega	Vrednosti spremenjivki izven zanke. Prav tako se bodo nadcin kako         lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo         pravilne.         45 minut         aktivno učenje, kolaborativno učenje, reševanje problemov         frontalni način podajanja snovi         individualno delo/delo v parih/skupinsko delo         (Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	Vrednosti spremenjivki izven zanke. Prav tako se bodo naučin kako lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne. 45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) Oskrbnica želi nahraniti mačke v desetih sobah mačjega zavetišča. V
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	Vrednosti spremenjivki izven zanke. Prav tako se bodo nadcin kako lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne. 45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) Oskrbnica želi nahraniti mačke v desetih sobah mačjega zavetišča. V vsaki sobi je naključno število posod (od 2 do 10), ki imajo različne
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	vrednosti spremenjoki izven zanke. Prav tako se bodo naučin kako lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne. 45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) Oskrbnica želi nahraniti mačke v desetih sobah mačjega zavetišča. V vsaki sobi je naključno število posod (od 2 do 10), ki imajo različne velikosti (od 1 do 5), posode, ki so v isti sobi pa se po velikosti ne
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	vrednosti spremenjivki izven zanke. Prav tako se bodo naučin kako lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne. 45 minut aktivno učenje, kolaborativno učenje, reševanje problemov frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo (Motivacija-uvod, Implementacija, Refleksija in vrednotenje) Oskrbnica želi nahraniti mačke v desetih sobah mačjega zavetišča. V vsaki sobi je naključno število posod (od 2 do 10), ki imajo različne velikosti (od 1 do 5), posode, ki so v isti sobi pa se po velikosti ne razlikujejo. Velikost posode pove koliko mačk lahko nahranimo iz nje,
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	<ul> <li>Iahko pridobijo uporabnikov odgovor, ga preverijo in štejejo pravilne.</li> <li>45 minut</li> <li>aktivno učenje, kolaborativno učenje, reševanje problemov</li> <li>frontalni način podajanja snovi individualno delo/delo v parih/skupinsko delo</li> <li>(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)</li> <li>Oskrbnica želi nahraniti mačke v desetih sobah mačjega zavetišča. V vsaki sobi je naključno število posod (od 2 do 10), ki imajo različne velikosti (od 1 do 5), posode, ki so v isti sobi pa se po velikosti ne razlikujejo. Velikost posode pove koliko mačk lahko nahranimo iz nje, npr. če je velikost posode 3, to pomeni, da bomo iz nje nahranili tri</li> </ul>
Trajanje aktivnosti Učne strategije in metode Učne oblike Povzetek učnega procesa	<ul> <li>vlednosti spremenjivki izven zanke. Prav tako se bodo naučili kako</li> <li>lahko pridobijo uporabnikov odgovor, ga preverijo in štejejo</li> <li>pravilne.</li> <li>45 minut</li> <li>aktivno učenje, kolaborativno učenje, reševanje problemov</li> <li>frontalni način podajanja snovi</li> <li>individualno delo/delo v parih/skupinsko delo</li> <li>(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)</li> <li>Oskrbnica želi nahraniti mačke v desetih sobah mačjega zavetišča. V</li> <li>vsaki sobi je naključno število posod (od 2 do 10), ki imajo različne</li> <li>velikosti (od 1 do 5), posode, ki so v isti sobi pa se po velikosti ne</li> <li>razlikujejo. Velikost posode pove koliko mačk lahko nahranimo iz nje,</li> <li>npr. če je velikost posode 3, to pomeni, da bomo iz nje nahranili tri</li> <li>mačke. Pomagaj ugotoviti število mačk, ki jih lahko nahranimo v</li> </ul>




posamezni sobi.

#### [1. korak]

Učencem naj na začetku izdelajo neko zanimivo ozadje za igro. Če želimo prihraniti čas, jim ga posredujemo v predlogi.



#### [2. korak]

Učenci naj privzetemu objektu (želvici) določijo nov izgled, ki bo predstavljal oskrbnico mačjega zavetišča.



#### [3. korak]

Preko pogovora, učencem pomagamo ugotoviti, da si bomo morali v igri uporabljati naslednje tri vrednosti in jih bomo morali zaradi tega hraniti v spremenljivkah: 1) število pravilnih odgovorov, 2) naključno število posod v eni sobi (2-10) in 3) naključno število, ki bo predstavljajo velikost posode znotraj ene sobe (1-5). Na začetku bomo morali nastaviti spremenljivko, ki bo hranila pravilne odgovore na 0, drugi dve pa bomo nastavili znotraj zanke, saj se bo njuna vrednost morala nastaviti pri vsaki ponovitvi. Le na ta način bomo za vsako sobo dobili nova naključna števila. Opozorimo jih na dejstvo, da za štetje sob ne bomo potrebovali dodatne spremenljivke, saj bomo v ta namen uporabili spremenljivko, ki jo uporablja zanka for za štetje ponovitev. Njena vrednost se bo namreč na začetku nastavila na 1, nato pa povečevala za ena do končne vrednosti: 10. Spreminjanje





vrednosti te spremenljivke v vsaki ponovitvi zanke pa nam ravno opisuje štetje sob. Na začetku, ko bomo v prvi sobi bo vrednost te spremenljivke 1, nato v naslednji, ko bomo v drugi sobi 2, in tako naprej do zadnje desete sobe.



#### [4. korak]

Na začetku igre moramo igralcu podati navodila za igranje. To naredimo z uporabo bloka "reci [besedilo] in počakaj [n] sekund", ki ga najdemo v skupini ukazov "Izgled". Učence opozorimo, da naj upoštevajo količino besedila, ki se bo prikazalo in ustrezno nastavimo čas prikazovanja.



#### [5. korak]

Z učenci se pogovorimo o tem, katere aktivnosti se bodo ponovile v vsaki sobi in bodo zato enake. Ukaze teh aktivnosti moramo postaviti znotraj telesa zanke, da se bodo izvedli pri vsaki ponovitvi te zanke. To kar se bo ponavljalo je naslednje: najprej bomo morali naključno določiti vrednost spremenljivke v kateri bomo hranili število posod, nato velikost posod, prav tako pa bomo morali vsakič uporabnika pozvati, da vpiše svoj odgovor. Odgovor bo treba nato vedno znova preveriti in podati ustrezno povratno informacijo v primeru ko bo odgovor pravilen oz. napačen. Ob pravilnem odgovoru bomo morali vrednost spremenljivke, ki nam bo služila za štetje pravilnih odgovorov, povečati za ena. Na tem mestu jim razložimo razliko med





blokoma "povečaj za" in "nastavi na". Pri prvem se bo trenutna vrednost povečala za izbrano vrednost, v drugem pa nastavila na neko vrednost. Za štetje bomo morali uporabiti prvo možnost.

#### [6. korak]

Spremenljivki naključno določimo vrednost z uporabo blokov: "nastavi [ime spremenljivke] na" in "naključno število od [n] do [m]".

nastavi stevilo\_posod na naključno število od 3 do 10 nastavi velikost\_posode na naključno število od 1 do 5

#### [7. korak]

Uporabnika bomo za vsako ponovitev pozvali, da naj vpiše število mačk, ki jih lahko nahranimo v posamezni sobi. Želimo, da se mu vprašanje izpiše tako, da bo v enem stavku dobil vse informacije (število posode, velikost posode), saj bo drugače posamezen podatek izpisan določeno število sekund, nato pa bo izginil, kar je lahko pri igranju moteče. To bomo lahko dosegli z združevanjem nizov, združen stavek pa bo tako kombinacija besedila in referenc na spremenljivke z vrednostmi. Nize lahko združimo z uporabo bloka "poveži", ki ga najdemo v skupini ukazov "Operatorji". Blok lahko razširimo tako, da pritiskamo na puščice na desni strani bloka, v konkretnem primeru bomo potrebovali pet prostorčkov.



#### [8. korak]

Združen niz nato postavimo v blok "vprašaj [besedilo] in čakaj", da na ta način pozovemo uporabnika za vnos in ga dobimo v program. Odgovor se bo shranil v blok "odgovor", ki ga najdemo v skupini ukazov "Zaznavanje".





Sofinancira program Evropske unije Erasmus+

poveži Stevilo posodica stevilo posod Avelikost posodica velikost posode in .Koliko muck lahko nahranim?

#### [9. korak]

Potem, ko igralec odgovori, moramo preveriti, če je odgovor pravilen. Učencem pojasnimo, da gre v tem primeru za situacijo z dvema možnostma, odgovor je namreč lahko ali pravilen, ali pa napačen. V teh situacijah pa uporabimo blok "če-sicer". Skupaj z njimi premislimo kako bomo ugotovili ali je odgovor pravilen. Pravilen odgovor se izračuna tako, da med seboj zmnožimo vrednosti: število posod in velikost posode. Če je igralčev odgovor enak tej vrednosti, potem je pravilen, drugače pa ne. V primeru, da je pravilen, moramo vrednost spremenljivke, kjer hranimo pravilne odgovore, povečati za ena in podati ustrezno povratno informacijo v obliki besedila. V primeru nepravilnega odgovora pa damo samo povratno informacijo, saj takrat vrednosti števca pravilnih odgovorov ne spreminjamo.



#### [10. korak]

V tem koraku se odločimo za ustrezno zanko. Če želimo šteti sobe, je najboljša možnost uporaba for zanke, saj lahko vrednost spremenljivke "i", ki šteje ponovitve zanke uporabimo pri štetju sob.

#### [11. korak]

Ko se izvajanje zanke zaključi je igre konec. Takrat posredujemo povzetek o uspešnosti igralca. Število pravilnih odgovorov imamo shranjeno v spremenljivki, število napačnih pa lahko izračunamo tako,





da vsem poskusom, ki jih je deset, odštejemo število pravilnih
odgovorov.
[Končna koda]
nastavi toke na D
reci Vizavetišču je 10 sob. za (3) sekund.
reci V'vsaki'sobi'sta'najmanj'3'in'največ'10'posodic'za'mucke. za 6 sekund.
reci Vserposodicervineki/sobi/sorenako/velike. za 4 sekund.
reci Vendarimajorrazlične soberrazlično velike posodice. za 4 sekund.
reci Velikost posodice je tanko od 1 do 0, za e sekund.
reci Čerjervelikost posodicer3, lahkorizn jernahranimornatančnor3 mucke, za (6) sekund.
reci Ugotovi koliko mucklahko nahranim v vsaki sobi. za (5) sekund.
for i = 1 to 10
nastavi stevilo_posod na naključno število od 3 do 10
nastavi velikost_posode v na naključno število od (1) do (5)
reci poveži Visobi 1 40 za 2 sekund.
Stevilo-posodic: stevilo_posod .Velikost-posode in
čakaj
če odgovor = stevilo_posod × velikost_posode
spremeni spremenljivko tocke 🗸 za 🚹
reci Odlično! Tvoj odgovor je pravilen, za (2) sekund.
reci Tołżałnipravilno, za (2) sekund.
reci
2 sekund.
če (i < 10)
reci Poskusi ugotoviti pravilno število v naslednji sobi. za (2) sekund.
reci Igrejekonec. za 2 sekund.
rect povez Pravino's cogovonia: tocke vkrat v za o sekund.
red povez Napacno'sroogovoria: 10 - tocke 'krat za 5 sekund.
UStavi vse





Orodja in viri za učitelja	<ul> <li>Celotna aktivnost v orodju Snap!: <u>https://snap.berkeley.edu/project?user=zapusek&amp;project=nah</u> <u>rani_mucke</u></li> </ul>
	<ul> <li>Lažja rešitev brez for zanke:</li> </ul>
	https://snap.berkeley.edu/project?user=zapusek&project=nah rani mucke spremenjeno
	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> </ul>
	<ul> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>
Viri/gradiva za	Predloga s slikovnim materialom:
učence	https://snap.berkeley.edu/project?user=zapusek&project=nah rani_mucke_predloga





## Učni scenarij 11 – Mačje zavetišče

Naslov učnega	Mačje zavetišče		
scenarija			
Pričakovano	<ul> <li>pogojni stavki (if blok)</li> </ul>		
programersko	<ul> <li>izpis besedila (blok reci)</li> </ul>		
predznanje			
Učni cilji	Splošni učni cilji:		
	<ul> <li>naključna števila</li> </ul>		
	<ul> <li>določanje vrednosti spremenljivkam</li> </ul>		
	uporabnikov vnos		
	<ul> <li>zanka "ponavljaj dokler"</li> </ul>		
	<ul> <li>pogojni stavki</li> </ul>		
	<ul> <li>operatorji za primerjanje</li> </ul>		
	• števec		
	Specifični učni cilji, ki so usmerjeni na algoritmično mišljenje:		
	<ul> <li>učenec zna prirediti vrednost spremenljivki,</li> </ul>		
	<ul> <li>učenec zna uporabiti blok "vprašaj [besedilo] in čakaj", da z njim pridobi uporabnikov vnos,</li> </ul>		
	<ul> <li>učenec zna uporabiti zanko "ponavljaj dokler", s katero ponavljajoče sprašuje uporabnika za vnos in nato vrednost vnosa tudi preveri,</li> </ul>		
	<ul> <li>učenec zna uporabiti pogojni stavek v povezavi z operatorji za primerjanje, da preveri pravilnost vnosa in poda ustrezen odziv,</li> </ul>		
	<ul> <li>učenec zna nastaviti izhodni pogoj zanke "ponavljaj dokler", da z njim preverja, če je igre konec,</li> </ul>		
	<ul> <li>učenec se zaveda, da ni potrebno posebej preverjati, če je igre konec, saj je preverjanje implicitno vključeno v pogoj zanke,</li> </ul>		
	<ul> <li>učenec zna implementirati števec, ki šteje ugibanja in uporabiti njegovo končno vrednost za razlikovanje med dvema različnima izidoma igre.</li> </ul>		





Cilji, naloge in kratek	Kratek opis: Izdelajte igro, v kateri bo igralec uganjeval naključno			
opis aktivnosti	določeno število od 1 do 100. Igralec bo uganjeval tako, da bo vpisoval			
	vrednosti in dobil povratno informacijo o tem ali je trenutno vpisano			
	število: večje, manjše ali enako številu, ki ga mora uganiti.			
	Naloga: Izdelajte igro, v kateri si bo oskrbnica zavetišča Marta na			
	začetku naključno določila število mačk v njenem zavetišču, vprašala			
	igralca po imenu in mu razložila navodila igre. Nato naj igralca			
	pozdravi z vpisanim imenom in začne s ponavljajočim spraševanjem o			
	številu mačk. Ko igralec vpiše svoj poskus, mu naj poda ustrezno			
	povratno informacijo: 1) če je vpisano število manjše, naj reče: "v			
	zavetišču je več mačk", 2) če je vpisano število večje, naj reče: "v			
	zavetišču je manj mačk", 3) če je vpisano število pravilno, pa:			
	"čestitke, uganili ste pravilno število mačk v zavetišču". V igro vključite			
	tudi števec poskusov, ko bo igralec število ugotovil pa preverite			
	njegovo vrednost. Če je vrednost števca manj ali enako 5, naj			
	oskrbnica reče: "Odlično ti je šlo, za nagrado si lahko izbereš eno.", v			
	drugem primeru pa: "Poskusi še enkrat."			
	Cilj: Učenci se bodo seznanili z zanko "ponavljaj dokler" in kako			
	lahko z nastavljanjem izhodnega pogoja ugotovijo, da je igre konec.			
	Prav tako se bodo naučili uporabljati spremenljivko v dveh različnih			
	situacijah: kot števec in za beleženje igralčevega vnosa.			
Trajanje aktivnosti	45 minut			
Učne strategije in	aktivno učenje, kolaborativno učenje, reševanje problemov			
metode				
Učne oblike	frontalni način podajanja snovi			
	individualno delo/delo v parih/skupinsko delo			





Povzetek učnega(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)procesaOskbnica mačjega zavetišča Marta želi, da uganeš število mačk, ki jih<br/>ima v svojem zavetišču. V zavetišču ima vedno vsaj eno mačko in nikoli<br/>več kot sto. Ko igralec vpiše svoj poskus mu pove, ali je vpisano<br/>število: manjše, večje ali enako uganjevanemu številu. Če igralec<br/>ugane število mačk v petih ali manj poskusih dobi za nagrado mačko,<br/>drugače pa ga oskrbnica pozove, da igro odigra še enkrat.

#### [1. korak]

Prva naloga, ki jo damo učencem je, da izdelajo zanimivo ozadje za igro. Učenci ga lahko narišejo sami, ali pa uporabijo slike iz spleta, pri čemer jih opozorimo oz. naučimo kako najdejo takšne, ki imajo ustrezno licenco. Če želimo prihraniti na času, jim sliko ozadja pripravimo vnaprej v predlogi.



#### [2. korak]

Učenci naj privzetemu objektu (želvici) določijo nov izgled, ki bo predstavljal oskrbnico mačjega zavetišča.







[3. korak]

Z učenci se pogovorimo, da bo igra bolj zanimiva za igranje, če se bo število mačk za vsako igro določilo naključno, saj bi drugače bila igra smiselna le enkrat. Naključno vrednost bomo uporabili pri primerjanju dejanske z vpisano vrednostjo, zato si jo moramo shraniti v spremenljivko. Spremenljivka je namreč (predpostavljamo, da še ne poznajo koncepta seznama) edini način, da si vrednost zapomnimo v programu Snap!. Določitev naključne vrednost se mora zgoditi preden začne igralec ugibati, skratka na začetku programa. Za ime spremenljivke izberemo nekaj mnemoničnega, da bomo že iz imena vedeli pomen vrednosti, ki je shranjena v njej.

#### ko kliknemo na

astavi stevilo\_mack na naključno število od 🚹 do 📶

#### [4. korak]

Skrbnica zavetišča vpraša igralca po imenu, zato da jo bo lahko pozdravila. To lahko naredimo z uporabo bloka *vprašaj [besedilo] in čakaj,* ki ga najdemo v skupini *Zaznavanje*. Igralčev odgovor se avtomatično shrani v spremenljivko *odgovor,* ki je ne ustvarimo sami, ampak je že privzeto v programu. S pomočjo odgovora, ki je shranjen v tej spremenljivki in bloka za povezovanje nizov združimo igralčevo ime s pozdravom. Za izpis besedila uporabimo blok *reci [besedilo] za [n] sekund.* Zaporedje teh blokov uporabimo tudi za podajanje navodil. Učence spomnimo, da je trajanje izpisa besedila povezano z njegovo dolžino in da je potrebno to smiselno nastaviti.







#### [5. korak]

Z učenci se pogovorimo, da ni mogoče predvideti kolikokrat bo igralec uganjeval, da bo ugotovil pravilno število. Lahko se mu posreči v prvem poskusu, lahko, da jih bo potreboval pet, tega ni mogoče predvideti. Pomembno je, da učenci ugotovijo, da gre pri tem problemu za situacijo, kjer potrebujejo ustrezno zanko. Tako, kjer bo njeno ponavljanje vezano na pogoj in ne na vnaprej določeno število ponovitev, t.j. zanka *ponavljaj, dokler <pogoj>*. Pogoj lahko relativno očiten, saj se bo igra izvajala dokler ne bo igralec ugotovil pravilnega števila oz. dokler ne bo število, ki ga je vpisal enako številu mačk.



#### [6. korak]

Nato moramo ugotoviti kateri ukazi se bodo ponavljali in jih bomo postavili v telo zanke. Torej kaj se bo ponavljalo dokler igralec ne bo ugotovil števila? Najprej ga bomo morali pozvati k ugibanju, nato pa se bomo morali glede na vpisano vrednost ustrezno odzvati.









#### [7. korak]

Zadnja stvar, ki se jo moramo z učenci pogovoriti je, kdaj bo zanka zaključila s ponavljanjem in kaj lahko iz tega sklepamo. Pa si podrobneje poglejmo kaj se bo zgodilo, ko bo igralec vpisal pravilen odgovor. Takrat bosta oba pogoja v telesu nepravilna, zato bo zanka šla v ponovno preverjanje pogoja, da se odloči ali se bo ponovila še enkrat. Pogoj v glavi zanke bo takrat izpolnjen, kar pomeni, da se bo zaključila. Ukazi, ki sledijo zanki se bodo tako izvedli le takrat, ko bo igralec vpisal pravilno število oz. povedano drugače, ko bo igre konec. Na to pa se ustrezno odzovemo.

#### [8. korak]

Če želimo šteti ugibe, moramo ustvariti spremenljivko v kateri bomo hranili to vrednost in jo na začetku programa nastaviti na 0. Z učenci se pogovorimo o pomenu prirejanja začetne vrednosti spremenljivke (inicializaciji) in o razliki med nastavljanjem in spreminjanjem vrednosti spremenljivke. V prvem primeru se vrednost nastavi na neko vrednost, prejšnja se na ta način izgubi, v drugem pa gre za relativno spreminjanje vrednosti, saj se trenutna vrednost spremeni za tisto, ki smo jo določili. V tej situciji bi radi števcu ugibov prištevali 1, za vsakič, ko bo igralec ugibal. Tako bomo morali zunaj zanke najprej nastaviti vrednost na 0, nato pa ga znotraj zanke povečevati za 1, ko bo na novo ugibal.





[9. korak]

lgralec dobi nagrado, če je uganjeval manj ali enako petkrat. To moramo preveriti po tem, ko je končal z uganjevanjem oz. takrat, ko se zanka zaključi. Snap! nima možnosti uporabe operatorja manj ali enako, zato uporabimo operator manj in preverjamo, če je bilo poskusov strogo manj od 6. Primer odločanja o nagradi je dober primer uporabe pogojne stavka if-else, saj imamo natanko dve možni situaciji - nagrado je dobil ali pa je ni dobil.

[Končna koda]















NAPREDNEJŠI UČNI SCENARIJI

## Učni scenarij 12 – Lovljenje zdrave hrane

Naslov	Lovljenje zdrave hrane	
učnega		
scenarija		
Pričakovano	Dodajanje besedila za lik	
programersk	Prikazovanje in skrivanje lika	
o predznanje	Uporaba ukaza obrni se v smeri	
	Uporaba naključnega števila	
	Uporaba spremenljivk za štetje točk	
	Uporaba zanke ponovi	
	Uporaba zanke za vedno	
	Uporaba pogojnega stavka	
Učni cilji	<ul> <li>Splošni učni cilji:</li> <li>Spremenljivke</li> <li>Pogojni stavek</li> <li>Zanka</li> <li>Obrni se v smeri</li> <li>Naključno število</li> </ul> Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje: <ul> <li>Učenec uporabi spremenljivko za preprečevanje prezgodnjega začetka</li> </ul>	
	<ul> <li>ocenice uporabi spremenjivko za preprecevanje prezgodnjegu začetka igre (opcijsko)</li> <li>Učenec zna uporabiti pogojni stavek (v kombinaciji s spremenljivko) za začetek igre</li> <li>Učenec uporabi <i>zanko ponovi dokler pogoj ni izpolnjen</i> za premikanje hrane</li> <li>Učenec uporabi <i>obrni se v smeri 180 stopinj</i> (dol) za premikanje likov navzdol</li> <li>Učenec uporabi naključno število za naključno hitrost premikanja likov hrane</li> <li>Učenec uporabi naključno število za pomikanje likov hrane na naključno pozicijo</li> <li>Učenec uporabi naključno število za pomikanje likov hrane na naključno x pozicijo in določeno y pozicijo (opcijsko)</li> </ul>	





I

Cilji, naloge	Kratek opis: Dekle lovi hrano tako, da klika z miško nanjo. Pri tem mora biti		
in kratek	pozorna, saj samo zdrava hrana prinaša točke!		
opis	Naloge: Učenci morajo sprogramirati dva različna lika: dekle, ki poda navodila,		
aktivnosti	in hrano, ki pada z vrha enkrana (vsa hrana ima skoraj enako kodo, le pri štetju		
	točk se malenkost razlikujeta kodi za zdravo in nezdravo hrano).		
	Dodatno lahko učenci dodajo spremenljivko in pogoj za nadziranje začetka igre.		
	Cilji: Učenci se bodo naučili premikanja za naključno število korakov,		
	premikanja na naključno pozicijo ter uporabe spremenljivke in pogojnega		
	stavka za preprečevanje dogodkov.		
Trajanje	45 min		
aktivnosti			
Učne	Aktivno učenje, učenje z izdelavo iger, reševanje problemov		
strategije in			
metode			
Učne oblike	Individualna učna oblika / Delo v paru		
Povzetek	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)		
učnega			
procesa	Dekle lovi hrano. Zdrava hrana prinese 1 točko, medtem ko nezdrava hrana		
	odšteje 1 točko. Na začetku se pojavi dekle, ki poda navodila. Nato se dekle		
	skrije in pojavi se hrana. Hrana pada iz vrha, dokler igralec ne doseže 5 točk. Na		
	koncu se ponovno pojavi dekle in zaključi igro.		





#### [Korak 1]

Aktivnost je namenjena individualnemu delu ali delu v paru. Učitelj pomaga z namigi, pojasni težje dele in po potrebi pomaga.

Na začetku je učencem na voljo:

- Ozadje
- Lik dekleta

Učenci lahko ozadje in lik dodajo / zamenjajo. Dekle na začetku poda navodila in nato izgine. Kot smo videli pri prejšnjih aktivnostih, moramo na začetku dekle najprej prikazati (če ostane skrita iz prejšnjih poskusov igranja). Primer kode:

ka k	liknemo na 🦰
prik	aži
reci	Živijo! za 👍 sekund.
reci	Pomagaj mi-ujeti-zdravo-hrano! za 4 sekund.
reci	Zdrava hrana prinese 1-točko, nezdrava pa-1, za (4) sekund.
reci	Igre je konec, ko dosežeš 5 točk. za 🚯 sekund.
reci	Pritisnitipko-S za začetek, za (2) sekund.
skrij	

H kodi dekleta se bomo še vrnili kasneje.

[Korak 2]

Dodamo nov lik zdrave hrane, npr. jabolko.

Najprej napišemo kodo za premikanje navzdol, zato uporabimo naslednja ukaza:



Ker ne želimo, da je jabolko narobe obrnjeno, kliknemo na ukaz *ne vrti*. Prav tako odkljukamo ukaz *vlečljiv*, saj to pomeni, da lika ne moremo premikati v miško (kar bi se med igro lahko zgodilo).



Da bo igra bolj zanimiva, nastavimo število korakov na naključno vrednost, kar pomeni, da bo hitrost premikanja poljubna, npr:











#### [Korak 5]

Na začetku ne želimo, da je jabolko prikazano, ampak želimo, da se prikaže, ko dekle konča z navodili, torej s klikom na tipko S. Uporabimo torej ukaz ko pritisnemo na tipko, nato se jabolko prikaže, ko dosežemo 5 točk pa se skrije. Koda trenutno izgleda tako:



#### [Korak 6]

Kaj se pa zgodi, ko kliknemo na jabolko?

Jabolko se mora skriti, prišteti moramo točko, nato se jabolko premakne na naključno pozicijo in ponovno prikaže. Število točk se bo povečalo za 1, koda za pozicijo pa je enaka prejšnji.



#### [Korak 7]

Vrnimo se h kodi za dekle.

Dekle se mora na koncu spet prikazati in reči npr. »Čestitam Ujel/a si dovolj zdrave hrane!« Za izvedbo le-tega bomo potrebovali *zanko za vedno*, ki bo konstantno preverjala, ali je igralec dosegel 5 točk. V primeru, da je igralec dosegel 5 točk, se dekle najprej prikaže, nekaj pove, nato pa izvajanje končamo z ukazom *ustavi vse* (brez tega ukaza bo dekle ponavljalo »Čestitam! ...« za vedno).

















	Jabolko		
	ko pritisnemo na tipko s de start = ] prikaži ponavljaj, dokler = tocke obrni se v smeri 180 premakni se naključno število od 1 de 2 korakov de se dotika rob 7 pojdi na x: naključno število od 200 y: 150 skrij strij		
	[Dodatne naloge]		
	Učenec doda dodatne naloge po svojih željah ali sledi spodnjim navodilom:		
Učni pripomočki, sredstva za učitelia	<ul> <li>Igro spremeni tako, da bo igralec hrano lovil s premikanjem sklede.</li> <li>Dodaj nov lik (skledo). Lik lahko narišeš, najdeš na spletu (pazi na pravice) ali pa uporabiš priloženo sliko sklede.</li> <li>Skledi nastavi ji začetno pozicijo (npr. na dnu ekrana) in naredi njeno premikanje levo in desno (po želji lahko tudi gor in dol). Živila morajo sedaj izginiti in se ponovno pojaviti na naključni lokaciji z dotikom sklede (in ne s klikom na živilo kot do sedaj).</li> <li>Spremeni pravila, da se igra konča, ko igralec doseže 20 točk (zmaga) ali ko pobere 3 nezdrava živila (izgubi).</li> <li>Dodaj več živil, da bo igra zanimivejša.</li> <li>Spremeni obleko liku skleda, ko igralec doseže npr. 5, 10, 15 točk.</li> <li>Primer celotne aktivnosti v Snap!-u: https://snap.berkeley.edu/project?user=mateja&amp;project=C4G12</li> <li>LovljenjeZdraveHrane</li> <li>Primer dodatne aktivnosti v Snap!-u:</li> </ul>		
učitelja	<ul> <li>Primer dodatne aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=mateja&amp;project=C4G12</u></li> </ul>		
	LovljenjeZdraveHrane%20%2B%20Dodatek		
	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>		
Učni	Predloga aktivnosti v Snap!-u:     https://www.barlade.com/units.com/un		
pripomočki za učenca	nttps://snap.berkeley.edu/project?user=mateja&project=C4G12_Lovije njeZdraveHrane - Delno Navedila za užensa (C4C12_NavedilaZal Jeensa deev)		
La uleilla	<ul> <li>Navodila za ucenca (C4G12_NavodilaZaUcenca.docx)</li> </ul>		





## Učni scenarij 13 – Sestavi zgodbo

Naslov učnega	Sestavi zgodbo	
scenarija		
Pričakovano	Prikazovanje in skrivanje lika	
programersko	Uporaba pogojnega stavka	
predznanje	Dodajanje besedila za lik	
	Uporaba ukaza	
	Menjava ozadja	
Učni cilji	<ul> <li>Splošni učni cilji:</li> <li>Premikanje lika in spreminjanje njegove velikosti</li> <li>Pošiljanje in prejemanje obvestil</li> <li>Struktra zgodbe</li> <li>Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:</li> </ul>	
	<ul> <li>Ocencec nactruje dialoge in aktivnosti likov v zgodbi</li> <li>Učencec pošilja obvestila za izvedbo dialogov med liki</li> <li>Učenec uporabi premikanje in spreminjanje velikosti likov</li> <li>Učenec uporabi skrivanje in prikazovanje lika</li> </ul>	
Cilji, naloge in	Kratek opis: Zajec pove zgodbo o Alici v	
kratek opis	čudežni deželi. Zgodba se začne v	
aktivnosti	gozdu. Alica hodi in se sprašuje »Kje sem?« /Za realizacijo njenega premikaja stran, se zmanjšuje velikost lika/. Alica pride do razpotja in tam vidi mačko na drevesu. Začne se pogovor med Alico in mačko. Pogovor je predsavljen na sliki. Naloge: Učenci eksperimentirajo s kratkim primerom iz zgodbe	
	(SestaviZgodboAlice1) o srečanju med Alico in mačko, ki temelji na	
	sinhronizaciji dialoga z uporabo ukaznega bloka <i>čakaj _ sekund</i> . Nato si	
	pogledajo drugo verzijo (SestaviZgodboAlice2) z uporabo pošiljanja	





obvestila. Učenci dokončajo zgodbo. Za pomoč jim je zgodba / scenarij ter		
napisani dialogi (v dveh tabelah na koncu dokumenta). Po želji lahko dodajo		
/ spreminjajo zgodbo.		
Cilji: Učenci se bodo naučili načrtovanja zgodbe, uporabe obvestil za		
sinhronizacijo aktivnosti likov in menjane ozadja.		
90 min		
Aktivno učenje, učenje z izdelavo iger, reševanje problemov		
Individualna učna oblika / Delo v paru / Skupinska učna oblika		
(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)		
Učitel/ica se z učenci pogovori o zgodbici Alice v čudežni deželi in jim pokaže		
sliko Alice, ki sreča mačko. Učencu si pozorno pogledajo kodo aktivnosti v		
Snap!-u - SestaviZgodboAlice1:		
https://snap.berkeley.edu/project?user=mateja&project=C4G13		
SestaviZgodboAlice1.		
Diskusija: Kdo spregovori prvi? Kdaj začne Alica in kdaj mačka? Zakaj ni		
sinhronizacije v dialogu? Odgovor leži v napačnem izračunu, kdaj kateri lik		
spregovori in kdaj naslednji nadaljuje z govorom.		
Skupaj si ogledamo kodo in tabelo:		
Macka		
Skripte Obleke Zvoki		
ko kiiknemo na		
pojdi na x: (74) y: (112) ko kliknemo na 🍋		
čakaj 10 sekund. pojdi na x: 187 y: 67 čakaj 9 sekund.		
reci Io-je-odvisno-od-tega, KAM-żeliś-priti! za 10 sekund. reci Živijo! za 2 sekund.		





Lik	Aktivnost	Kdaj	Kdaj	Trajanje
		začne	konča	
Zajec	Reče: SI že slišal/a za Alice v	0	14	14
	čudežni deželi? Poglejmo si			
	eno od njenih zgodbic!			
Alica	Reče: "Mi lahko prosim	9	21	12
	poveš, kam moram iti?"			
Mačka	Reče: "To je odvisno od tega,	10	20	10
	KAM želiš priti!"			

Kot lahko vidimo, je sinhronizacija dialogov z uporabo ukaza *čakaj \_\_\_\_ sekund* zelo zakomplicirana in lahko vodi do številnih napak v zgodbi.

Učitelj/ica pokaže nekoliko drugačno kodo aktivnosti v Snap!-u – SestaviZgodboAlice2:

https://snap.berkeley.edu/project?user=mateja&project=C4G13 SestaviZgodboAlice2.











ter vpišemo ime obvestila.



Uporaba dogodka:

Izberemo ukazni blok *ko sprejmem* \_\_\_, kliknemo na puščico in izberemo željeno obvestilo.

Kar zapišemo pod ta ukazni blok se bo izvedlo, ko bo sporočilo prejeto.

Učenci v parih dokončajo zgodbo z začetne slike. Obvestilo, ki ga mačka pošlje Alici, npr. omenujejo Alice2, obvestilo, ki ga Alica pošlje mački, pa macka1.

Učitelj/ica pripomne, da pripovedovanje zgodb navadno vsebuje tudi menjavo ozadij. »Dopolnimo zgodobo tako, da zgodbo začne Zajec na ozadju start, nato se ozadje spremeni (gozd), pojavi se Alica, se sprašuje »Kje sem?« ter se oddaljuje in izginja v gozd, nato se ozadje zamenja (v ozadje srecanje), pojavi se Macka in začne se pogovor med Alico in Macko.«

Učitelj/ica pokaže celotno zgodbo – SestaviZgodboAlice: <u>https://snap.berkeley.edu/project?user=mateja&project=C4G13</u> <u>SestaviZgodboAlice</u>.

Skupaj komentiramo zgodbo.

»Kdaj se zamenja ozadje? Kdaj se Alica prikaže in kaj počne? Kdaj se pojavi Macka in kaj počne ona?«

Spet si ogledamo celotno aktivnost Alice. Imamo 3 ozadja, od tega je eno v uporabi. S katerim ozadjem začnemo? Kaj moramo narediti, da se Alica in





Macka ne pojavita na začetku? Kako zamenjamo ozadje? Ozadje lahko zamenjamo s pošiljanjem obvestila in sicer ko Zajec konča z uvodom. Alica se pojavi, ko se ozadje zamenja z obvestilom *Pojdi v gozd*.

#### **Opozorilo**:

Končna aktivnost SestaviZgodboAlice (leva slika) se od delne aktivnosti SestaviZgodboAlice 2 (desna slika) na začetku razlikuje, zato je tudi prvo obvestilo, ki ga pošlje Zajec, nekoliko drugačno, saj je dodana scena z gozdom:



# ko kliknemo na **Na**reci Živijoj za 🔁 sekund.

reci Słżesliśal/aza Alicewcudeżni deżeli? za 6 sekund. reci Pogiejmosi enorod njenih zgodbici za 6 sekund. pošilji Alicei vsem





Sofinancira program Evropske unije Erasmus+







	ni prikazana, saj je del ozadja. Prikaže se z obvestilom macka1.			
	Povemo lahko, da je bil lik mačke izrezan iz ozadja z drugim programom,			
	kajti Snap! tega na žalost ne omogoča.			
	Ko zajec konča in pošljemo obvestilo Alice1, se zgodba nadaljuje kot v			
	aktivnosti SestaviZgodboAlice2.			
	Učitelj/ica komentira, da moramo pri pripovedovanju in sestavljanju zgodb			
	najprej izdelati scenarij. V nadaljevanju se nahaja tabela, ki bo pri tem v			
	pomoč. Tabelo lahko po želji spremeni / doda scene učitelj ali pa učenci.			
	Učenci v parih dokončajo scenarij in zgodbo v Snap!-u. Nadaljujejo z			
	aktivnostjo SestaviZgodboAlice2.			
Učni	Primer celotne aktivnosti v Snap!-u:			
pripomočki,	https://snap.berkeley.edu/project?user=mateja&project=C4G13			
sredstva za	<u>SestaviZgodboAlice</u>			
učitelja	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani</li> <li>računalnički mačak, Liubliana: Pasadona</li> </ul>			
	<ul> <li>Vorderman, C. (2017). Računalniško programiranje za otroke.</li> </ul>			
	Ljubljana: MK.			
Učni	<ul> <li>Začetni primer v Snap!-u: https://snap.borkolov.edu/project?usor=mateia&amp;project=C4C12</li> </ul>			
pripomočki za	nups://snap.berkeiey.edu/project?user=mateja&project=C4G13			
učenca	Sestavi2godboAlice1			
	<ul> <li>Predioga aktivnosti v Snap!-u: https://snap.berkeley.edu/project?user=mateja&amp;project=C4G13</li> </ul>			
	SestaviZgodboAlice2			
	<ul> <li>Navodila za učenca, ki vsebuje tudi spodnji tabeli z</li> </ul>			
	zgodbo/scenarijem in liki (C4G13_NavodilaZaUcenca.docx)			

### Zgodba / scenariji

Ime	Ozadje	Akcija	Notes
		103	





1. start	Alice in the Wonderland	Zgodba se začne s	Na tem ozadju Zajec predstavi
		prizorom (ko	zgodbo.
		kliknemo na zeleno	
		zastavico).	
2. gozd		Ta scena se pojavi,	Alica se pojavi na sredini
		ko Zajec pove	zaslona. Začne se premikati in se
		uvodni del (ko	sprašuje » <i>Kje sem?</i> «. Lik
		program sprejme	postopoma zmanjšuje svojo
		obvestilo <i>Pojdi v</i>	velikost (5x za -10). Ko pride do
		gozd).	konca poti (do razpotja), se
			scena spremeni na <i>srecanje</i> .
			Alica pošlje obvestilo Srecanje z
			macko.
3. srecan je		Ta scena se pojavi,	Ko se prizor spreminja, Zajec
		ko program sprejme	nadaljuje s pripovedovanjem
		obvestilo <i>Srecanje z</i>	zgodbe. Kasneje poteka pogovor
		macko.	med Alico in Macko.



Liki



Lik	Akcija	Ozadje	
and the second se	Na začetku:	Alice in the Wonderland	
N.	Reče: Živijo! (za 2 sek.)		
	Reče: SI že slišal/a za Alico v čudežni deželi? (za 6 sek.)		
	Reče: Poglejmo si eno od njenih zgodbic! (for 6 sec.)	start	
	Pošlje obvestilo <i>Pojdi v gozd</i> .		
×	Na začetku:	Alsee	
April 1	Se skrije.		
	Pripravimo jo na novo ozadje (gozd), tako da gre na x: 0, y: 0 in	100	
	nastavi velikost na 100%.	start	
	Na začetku:	Alsce	
	Se skrije.	in the wooder and	
	Pripravimo jo na novo ozadje ( <i>srecanje</i> ), zato gre na x: -74, y:		
	113.	start	
	Prejme obvestilo <i>Pojdi v gozd</i> :	124 15 3	
Albert.	Lik se prikaže.	A How	
	Obrne se v smeri 0.		
	5x se ponovi: čakaj 1 sekundo, premakni se 5 korakov, spremeni	gozd	
	velikost za -10, misli si » <i>Kje sem?</i> «.		
	Priprava na novo ozadje: čakaj 5 sekund, nastavi velikost na 100%		
	in pojdi na x: -187, y: -67.		
	Pošlje obvestilo <i>Srecanje z macko</i> .		
N.	Brez akcije. Ostane prikazan od prej.		
		gozd	
142	Prejme obvestilo <i>Srecanje z macko</i> .	1	
No.	Nastavi veliksot na 80%		
	Reče: "Alica se ustavi na razpotju in sprašuje, kam naj gre." (za 10	1	





	sek.).	srecanje	
	Reče: <i>" Nato vidi mačko na drevesu."</i> (za 8 sek.)		
	Pošlje obvestilo Alice1		
<b>A</b>	Prejme obvestilo Alice1.		
	Postavi se v ospredje - blok go to front layer (drugače se lahko		
	zgodi, da se lik Macke pojavi čez oblak in ni vidno, kaj Alica		
	govori).	srecanje	
	Reče: "Živijo!" (za 2 sek.)		
	Reče: "Mi lahko prosim poveš, kam moram iti?" (for 10 seconds).		
	Pošlje Macki obvestilo macka1.		
	Prejme obvestilo macka1.	and the second second	
	Lik Macke se pojavi na zaslonu.		
	Reče: " To je odvisno od tega, KAM želiš priti!" (za 10 sekund).		
	Pošlje obvestilo <i>Alice2</i> .	srecanje	
	Prejme obvestilo <i>Alice2</i> .	and the second second	
aber .	Reče:	A Call	
	Pošlje obvestilo <i>macka2</i> .	4	
		srecanje	
	Prejme obvestilo macka2.		
	Reče:		
	Pošlje obvestilo <i>zajec1</i> .	100 miles	
		srecanje	
1	Prejme obvestilo <i>zajec1</i> .		
S.	Reče: "Kaj je morala zgodbe?" (za 8 sek.)		
	Reče: "Če želiš vedeti, kam naprej, moraš najprej določiti svoj		
	cilj."	srecanje	

## Učni scenarij 14 – Onesnažen zrak





Naslov učnega	Onesnažen zrak
scenarija	
Pričakovano	Učenci poznajo in znajo uporabljati bloke za risanje, spreminjati izgled
programersko	lika, dodajati nove spremenljivke, uporabljati zanke, uporabljati
predznanje	računske operatorje.
Učni cilji	Splošni učni cilji:
	<ul> <li>Učenci se spoznajo s konceptom paralelizma v orodju Snap!</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>učenci uporabljajo bloke za risanje;</li> <li>učenci uporabljajo zanke za risanje likov (kvadrat, krog);</li> </ul>
	<ul> <li>učenci uporabijo spremenljivko za štetje točk;</li> </ul>
	<ul> <li>učenci uporabijo naključna števila za določanje pozicije svinčnika;</li> </ul>
	<ul> <li>učenci uporabijo sporočila za paralelno izvajanje dogodkov.</li> </ul>
Cilji, naloge in kratek	Kratek opis: Učenci sestavijo igro, katere cilj je očistiti onesnažen zrak
opis aktivnosti	v okolici industrije. Igralec onesnažen zrak poskuša izboljšati s
	posaditvijo dreves (smrek in hrastov), pri čemer zbira točke. Ob
	narisani smreki prejme 2 točki, za narisan hrast pa 3 točke, saj je ta
	bolj učinkovit pri čiščenju zraka, kot smreka. Igralec ima možnost
	narisana drevesa izbrisati. V tem primeru se točke izničijo. Po zbranem
	določenem številu točk (npr. 10) se zrak v okolju industrije izboljša in
	posledično prične rast zelenje. Ko zrak več ni onesnažen, se igra
	zaključi.
	Naloga: Učenci naprej definirajo začetek igre, v kateri je v ozadju
	industrija in na sredini ekrana svinčnik. Nato napišejo kodo za risanje
	smreke, ki se izriše v primeru, ko igralec klikne na ikono smreke.
	Podobno sestavi kodo tudi za izris hrasta. V naslednjem koraku učenec
	sestavi kodo za možnost izbrisa vseh narisanih dreves. Nato doda še
	točkovanje, kjer igralec za vsako narisano smreko prejme 2 točki, za
	vsak hrast pa 3. Učenec mora nato določiti, koliko točk mora igralec





	doseči (npr. 10), da se igra zaključi. Učenec mora dodati kodo tako, da	
	se ozadje spremeni, ko igralec doseže določeno število točk.	
	Cilji: Učenci se spoznajo sm konceptom paralelizma. Učenci razumejo	
	v skripte katerih likov je potrebno dodajati kodo. Za paralelno	
	izvajanje dogodkov v skriptah različnih likov uporabljajo boke za	
	pošiljanje in sprejemanje sporočil.	
Trajanje aktivnosti	45 minut	
Učne strategije in	učenje z igro, učenje z ustvarjanjem igre, problemsko učenje	
metode		
Učne oblike	Frontalna oblika, delo v paru	
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)	
procesa		
	Motivacija-uvod	
	Učencem podamo v igranje že ustvarjeno igro, da postanejo	
	motivirani za ustvarjanje svoje. Ob igri se pogovorimo o konceptih, ki	
	jih bomo spoznali skozi sestavljanjem igre. Na primeru jim frontalno	
	pokažemo uporabo blokov za pošiljanje in sprejemanj sporočil. Učenci	
	lahko predlagajo nadgradnjo igre, ki jo lahko tudi realizirajo na koncu	
	šolske ure.	
	Implementacija	
	<ol> <li>Učenci odprejo predlogo programa, v kateri sta dodani dve ozadji – industrija in travnik ter liki, kot so svinčnik za risanje dreves, križec z namenom brisanja narisanih dreves, ikoni smreke in hrasta.</li> </ol>	










smreko, kot je prikazana na sliki.



Pri risanju uporabljamo bloke svinčnika. Da svinčnik spušča za sabo sled, je potrebno najprej svinščnik spustit. Nato dodamo kodo za risanje krošnje, kjer je potrebno najprej nastaviti barvo svinčnika in smer svinčnika v desno stran.

> svinčnik spuščen nastavi barvo svinčnika na obrni se v smeri 90 🗸

Sledi risanje krošnje v obliki trikotnika. S pomočjo zanke narišemo tri stranice trikotnika tako, da se svinčnik najprej pomakne za 40 korakov (lahko več ali manj, odvisno kako velike krošnje želimo) in nato obrne v levo za 120° (priležni kot notranjega kota enakostraničnega trikotnika). To ponovi 3krat.



Sledi risanje debla v obliki kvadrata. Ker je svinčnik končal z risanjem v levem kotu, ga je potrebno premakniti za 25 korakov (da bo deblo na sredini - stranica je dolga 40 korakov, 10 korakov pa bo široko deblo) v desno, kjer bo začel z risanjem debla. Svinčniku nastavimo rjavo barvo.

> premakni se (25) korakov nastavi barvo svinčnika na

Potem z zanko narišemo tri stranice kvadrata (4. stranico ni potrebno risati po krošnji) tako, da svinčnik najprej obrnemo za 90° v desno in nato premaknemo za 10 korakov, kar se











svinčnik spuščen nastavi barvo svinčnika na obrni se v smeri 90 -

Sledi risanje krošnje v obliki kroga. Krog narišemo s pomočjo zanke. V tem primeru se zanka ponovi 120-krat, pri čemer se svničnik pomakne za 1 korak in se obrne v levo za 3° (podatki so lahko tudi drugačni, pri tem je potrebno le upoštevati, da se izriše polni krog 360°).



Sledi risanje debla v obliki kvadrata, po podobnem postopku kot pri risanju smreke. V kodi določimo, kje naj začne z risanjem debla (premik za npr. 3 korake) in svinčniku nastavimo rjavo barvo.

> premakni se 3 korakov nastavi barvo svinčnika na

Nato dodamo isto kodo kot pri risanju smreke. Narišemo deblo v obliki kvadrata, prekinemo risanje z dvigom svinčnika in naključno izberemo novo lokacijo svinčnika.



5. Naslednji korak je dodajanje možnosti izbrisa vseh narisanih dreves. Ko je z miško kliknjen križec (lik *Počisti*), ta svinčniku pošlje sporočilo, naj počisti vsa narisana drevesa.























	Ko je miška kliknjena – pošlji počisti – vsem
Učni pripomočki,	Primer celotne aktivnosti v Snap!-u:
sredstva za učitelja	https://shap.berkeley.edu/project?user=tadeja&project=Ones naževanje zraka
	• Lajovic, S. (2011). Scratch. Nauči se programirati in postani
	računalniški maček. Ljubljana: Pasadena.
	<ul> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>
Učni pripomočki za	<ul> <li>Predloga aktivnosti v Snap!-u:</li> </ul>
učenca	https://snap.berkeley.edu/project?user=tadeja&project=Zrak za%C4%8Detno
	<ul> <li>Navodila za učenca (C4G14_NavodilaZaUcenca.docx)</li> </ul>





## Učni scenarij 15 – Ulovi miš

Naslov učnega	Ulovi miš
scenarija	
Pričakovano	<ul> <li>dodajanje ozadja;</li> <li>dodajanje povoga likaj</li> </ul>
programersko	<ul> <li>dodajanje novega ika,</li> <li>dodajanje zvoka;</li> </ul>
predznanje	• govorjenje lika;
	<ul> <li>spreminjanje obleke lika kot animacija;</li> </ul>
	<ul> <li>premikanje lika s smernimi tipkami;</li> </ul>
	• pogojni stavki.
Učni cilji	Splošni učni cilji:
	<ul> <li>spoznavanje neskončne zanke;</li> </ul>
	<ul> <li>spoznavanje naključnih števil;</li> </ul>
	<ul> <li>spoznavanje števnika;</li> </ul>
	<ul> <li>spoznavanje časovnika;</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>učenci uporabijo neskončno zanko za premikanje lika;</li> <li>učenci uporabijo naključna števila za določanje naključne pozicije lika, premikanje lika za naključno število korakov in obračanje lika za naključno število stopinj;</li> </ul>
	<ul> <li>učenci ustvarijo števnik z dodajanjem nove spremenljivke in končni rezultat uporabijo za povzemanje uspešnosti igralca pri lovljenju miši;</li> </ul>
	<ul> <li>učenci uporabijo časovnik za določitev konca igre.</li> </ul>
Cilji, naloge in kratek	Kratek opis: Učenci sestavijo igro, v kateri igralec s premikanjem
opis aktivnosti	mačke lovi miš.
	Naloga: Naloga učenca je, da sestavi igro lovljenje miši. Mačko
	premika igralec s smernimi tipkami, miš pa se po sobi premika
	naključno. Miš vedno mački uide, zato vsakič, ko se mačka dotakne
	miši, se ta prikaže na naključni lokaciji v sobi. Potrebno je dodati
	števnik, ki šteje kolikokrat se mačka dotakne miši. Igra se zaključi po
	določenem času (nnr. 30. s), zato notrebujemo še časovnik. Na koncu





	igre, dekle pove kolikokrat je mačka ujela miš.
	<b>Cilj:</b> Učenec se spozna z dodeljevanjem naključnih vrednosti in pri tem
	se nauči kako uporabiti operator <i>naključno_število_od_[x]_do_[y]</i> .
Trajanje aktivnosti	45 min
Učne strategije in	Aktivni pouk, sodelovalno učenje, problemsko učenje, učenje z
metode	ustvarjanjem iger
Učne oblike	Frontalna oblika
	Delo v parih ali v manjših skupinah
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	Motivacija-uvod
	Učence motiviramo s prikazom igre, ki jo bodo morali tudi sami
	sestaviti. Ob igri se pogovorimo, kako začeti z ustvarjanjem igre.
	Skupaj z učenci oblikujemo korake ustvarjanja igre, kot na primer:
	<ol> <li>izberi ozadje in dodaj like;</li> <li>mački sestavi program tako, da se bo premikala s smernimi tipkami;</li> <li>miši sestavi program tako, da se bo premikala naključno;</li> <li>sestavi program tako, da se bo miš ob vsakem dotiku mačke skrila in prikazala na naključni lokaciji v sobi;</li> <li>dodaj števnik, ki bo štel kolikokrat se mačka dodakne miš;</li> <li>dodaj časovnik in določi konec igre;</li> <li>dodaj dekle, ki na koncu igre pove, kolikorat je mačka ujela miš;</li> <li>dekletu sestavi program tako, da bo skočila vsakič, ko se dotakne miši;</li> <li>dodaj zvok miši ali mačke;</li> <li>itd.</li> </ol>





določanje naključne vrednosti: naključno število od (1) do (10) Učenci nato pričnejo z ustvarjanjem igre v parih/manjših skupinah. Učitelj jim je pri tem v podporo in pomoč. Implementacija [1. korak] Prvi korak je dodajanje ozadja. Učenci lahko na spletu poiščejo sliko, ki je prosto dostopna in ima dovoljenje za uporabo. V naslednjem koraku dodajo nova lika – mačko in miš. [2. korak] V drugem koraku učenci sestavijo program tako, da se mačka premika s smernimi tipkami. Določiti morajo tudi, kaj se zgodi, če mačka pride na rob ekrana. to kilknemo na prikaži nastavi velikost na 🔞 % če tipka puščica gor pritisnjena? premakni se 10 korakov obrni se v smeri 💽 če tipka publicadol - pritisnjena? premakni se 10 korakov obrni se v smeri 180 🕶 če tipka puśćice desno - pritisnjena? premakni se 10 korakov obrni se v smeri (90 🗸 če tipka puščica levo – pritisnjena? premakni se 10 korakov obrni se v smeri -90 🕶 odbij se, če si na robu [3. korak]





V naslednjem koraku je potrebno sestaviti program tako, da se miš po sobi premika naključno. V tem primeru je koda sestavljena tako, da miš neskončnokrat (zanka za vedno) premakne za naključno število korakov in se obrne za naključno število stopinj. V program je tako potrebno dodati bloka *Premikanje/premakni\_se\_[x]\_korakov* in *Premikanje/obrni\_se\_[x]\_stopinj*, v katera vstavimo operator *naključno\_število\_od\_[x]\_do\_[y]*.



#### [4. korak]

Sedaj je potrebno sestaviti program tako, da se miš ob dotiku mačke skrije in prikaže na naključni lokaciji v sobi. V tem primeru miš pobegne mački. Učenci lahko tukaj dodajo tudi svoje pravilo igre. V vsakem primeru pa morajo v programu uporabiti operator *naključno\_število\_od\_[x]\_do\_[y]*.



#### [5. korak]

V tem koraku dodamo igri števec, ki bo štel kolikorat mačka ulovi miš oz. kolikokrat se mačka dotakne miši. Učenci morajo najprej ustvariti novo spremeljivko, v našem primeru poimenovano *Rezultat*, in jo dodati v kodo mačke. Pri tem je potrebno paziti, da je spremenljivka *Rezultat* na začetku igre nastavljena na vrednost 0. To naredimo s pomočjo bloka *Spremenljivke/nastavi\_[spremenljivka]\_na\_[x]*. V primeru prikazovanja vrednosti spremenljivke *Rezultat* na igralnem ekranu dodamo v kodo blok

















Učenci prilagodijo svojo kodo: mačka se za vedno premika naključno od 20 do 60 korakov; ٠ ko se miš dotakne mačke, gre na naključno lokacijo, pri čemer ٠ je x = 100; miš se vedno obrne za 90 stopnij; ipd. [Končna koda] Mačka ko kliknemo na 🍋 Ko je se dotika Miš - ? prikaži spremeni spremenljivko Rezultat = za 🕕 nastavi velikost na 60 % predvajaj zvok Meow nastavi Rezultat - na 0 čakaj 1 sekund. prikaži spremenljivko Rezultat 👻 za vedno če tipka puščica gor - pritisnjena? Ko je štoparica = 30 premakni se 10 korakov skrij obrni se v smeri 🛛 🔻 ustavi vse zvoke če tipka puščica dol v pritisnjena? skrij spremenljivko Rezultat premakni se 10 korakov obrni se v smeri 180 🗸 če tipka puščica desno - pritisnjena? premakni se 10 korakov obrni se v smeri 🧕 🗸 če tipka puščica levo - pritisnjena? premakni se 10 korakov obrni se v smeri 🕞 🗣 odbij se, če si na robu Miš



	ko kliknemo na prikaži za vedno če se dotika Mačka ? pojdi na x: naktjučno število od _200 do 200 y: naktjučno število od _200 do 200 premakni se naktjučno število od 10 do 100 korakov odbij se, če si na robu čakaj 0.35 sekund. obrni se Č naktjučno število od 20 do 90 stopinj odbij se, če si na robu
	Dekle
	ko kliknemo na       ko je štoparica = 30         pojdi na x: -9 y: 100       nastavi velikost na 150 %         nastavi velikost na 100 %       nastavi velikost na 150 %         prikaži       ce se dotika Mis ?         Preklopi na obleko ballerina d sicer       reci Nisiruje/arnobenermišil za 3 sekund.         Preklopi na obleko ballerina a moleko ballerin
	Ozadje ko kliknemo na Re reset štoparice
Učni pripomočki,	<ul> <li>Primer celotne aktivnosti v Snap!-u: https://snap.berkeley.edu/project?user=tadeja&amp;project=</li> </ul>
sredstva za učitelja	Ulovi%20mi%C5%A1_1
	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>
Učni pripomočki za	<ul> <li>Predloga aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=tadeja&amp;project=</u></li> </ul>
	Ulovi%20mi%C5%A1





## Učni scenarij 16 – Kupovanje hrane za piknik

Naslov	Kupovanje hrane za piknik
učnega	
scenarija	
Pričakovano	Dodajanje besedila za lik
programersk	Prikazovanje in skrivanje lika
o predznanje	Uporaba operatorjev
	Uporaba spremenljivk
	Uporaba združevanja nizov
	Uporaba pogojnega stavka
Učni cilji	Splošni učni cilji: <ul> <li>Spremenljivke</li> <li>Pogojni stavek</li> <li>Operatorji</li> </ul>
	<ul> <li>Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje: <ul> <li>Učenec uporablja spremenljivke za določanje cene izdelka</li> <li>Učenec prireja vrednosti spremenljivke glede na porabo denarja</li> <li>Učenec uporablja pogojni stavek za preverjanje količine denarja, ki je na voljo</li> <li>Učenec uporablja operatorje za združevanje nizov (besedilo – vrednost spremenljivke – besedilo)</li> <li>Učenec uporablja operatorje za primerjavo cene izdelka in denarja, ki je na voljo</li> <li>Učenec uporablja operatorje za spreminjanje (odštevanje) vrednosti spremenljivke</li> </ul> </li> </ul>
Cilji, naloge	Kratek opis: Dekle potebuje pomoč pri kupovanju hrane za piknik. Na voljo ima 15
in kratek opis	EUR. Ko kupi določen izdelek, se vrednost njene »denarnice« (denarja, ki ga ima
aktivnosti	na voljo) spremeni. Če je v denarnici premalo denarja, izdelka ne more kupiti.
	Naloge: Učenci morajo napisati tri različne kode: za dekle, hrano (lik podvojijo z
	majhno spremembo v kodi) ter gumbom za konec igre. Dekle pove navodila in
	koliko denarja je na voljo, s klikom na gumb konec pa pove še, koliko zdravih in
	nezdravih izdelkov je igralec kupil. Ko gremo z miško čez določen izdelek, se izpiše
	njegova cena. Če ima igralec na voljo dovolj denarja, lahko izdelek kupi, v
	nasprotnem primeru pa ga ne more kupiti.





	Cilji: Učenci se bodo naučili delati s spremenljivkami: nastavljanje različne
	začetne vrednosti, uporaba pogojnega stavka za primerjanje vrednosti
	spremenljivk, spreminjanje vrednosti spremenljivk z odštevanjem, uporaba
	spremenljivk za štetje (ne)zdrave hrane. Dodatno, ponovili bodo dodajanje
	besedila, združevanje besedila in pogojni stavek.
Trajanje	45 min
aktivnosti	
Učne	Aktivno učenje, učenje z izdelavo iger, reševanje problemov
strategije in	
metode	
Učne oblike	Individualna učna oblika / Delo v paru
Povzetek	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
učnega	
procesa	Dekle v trgovini nakupuje hrano za piknik. Na voljo ima 15 EUR. Ceno izdelka
	lahko vidi, ko se z miško pomakne čez izdelek, kupi pa ga s klikom na izdelek. Kupi
	ga lahko le, če ima na voljo dovolj denarja. S klikom na gumb Konec pove, koliko
	zdravih in nezdravih izdelkov je bilo kupljenih.
	Torta stane 8 EUR.
	KONEC
	[Korak 1]

















Možna koda:
Ko je miška kliknjena - če (cenaLubenice > denarnica) reci Nimašdovojidenarjal za 5 sekund. sicer reci Odlična izbiral za 2 sekund. nastavi denamica = na (denarnica = cenaLubenice) spremeni spremenijivko zdravaHrana = za 1 skrij
[Korak 4]
Na polici želimo imeti več izdelkov, zato lik lubenice podvojimo. Drugi lik bo npr.
torta. Koda iz [Korak 3] potrebuje nekaj sprememb in sicer moramo:
<ul> <li>zamenjati obleko,</li> <li>ustvariti novo spremenljivko: <i>cenaTorte</i>,</li> <li>nastaviti ceno torte = nastaviti vrednost spremenljivke <i>cenaTorte</i>,</li> <li>v kodi zamenjati spremenljivko <i>cenaLubenice</i> v <i>cenaTorte</i>,</li> <li>zamenjati spremenljivko <i>zdravaHrana</i> v <i>nezdravaHrana</i>.</li> </ul>
Koda za torto, ko je miška kliknjena, bo npr.: Ko je miška kliknjena če ( cenaTorte > denamica [recl Nimaš dovo];denarja] za (5) sekund. siccr recl Preveč;sladkorja] za (2) sekund. nastavi denamice na denamica ( cenaTorte ) spremeni spremenijivko nezdravaHrana za (1)
Korak večkrat ponovimo, da imamo na polici več izdelkov.
[Korak 5] Ko igralec konča z nakupovanjem, klikne na lik Konec. Lik konec pošlje obvestilo o koncu nakupovanja. [Korak 6]





Vrnimo se h kodi dekleta.
Sedaj želimo, da dekle pove, koliko zdravih in nezdravih izdelkov je igralec kupil.
Uporabimo dogodek <i>ko sprejmem</i> obvestilo <i>konec</i> :
ko sprejmem konec - reci poveži Izbral/arsiz zdravaHrana izdravih živikim nezdravaHrana inazdravih živil () 20 5 sekund.
[Korak 7]
Kadarkoli med igro lahko igralec preveri, koliko denarja ima še na voljo. To stori
tako, da se z miško pomakne na dekle. Primer kode:
Ko je miška se dotika <del>–</del> reci poveži imaši denarnica (EUR, 4) za (2) sekund.
[Končna koda]
Dekle
ka kliknemo na nastavi denamica = na 15 nastavi zdravaHrana = na 0 nastavi nezdravaHrana = na 0 reci Živijol za 2 sekund. reci Pomaga mi-kupiti-hrano:za-pikniki za 4 sekund. ko sprejmem konec = reci poveži Izbral/a-si* zdravaHrana *zdravih-živil-in* nezdravaHrana *nazdravih-živil. (* <sup>24</sup>
Ko je miška se dotika <b>*</b> reci poveži imaši denarnica iEUR. () za (2 sekund.
Hrana





	ko kilknemo na prikaži nastavi centi Jagode na č Ko je miška se dotika * misli poveži Jagode stanejo: cena Jagode *EUR, ** za 2 sekund Ko je miška kilknjena * če cena Jagode > denarnica reci Nimaš dovolj denarjal za š sekund. sicer reci Odlična izbiral za 2 sekund. nastavi denamka = na denarnica = cena Jagode spremeni spremenijivko zdravalirana za 1
	Копес
	Ko je miška kliknjena – pošiji konec – vsem
	[Dodatne naloge]
	Učenec doda dodatne naloge po svojih željah ali sledi spodnjim navodilom:
	<ul> <li>Spremeni igro tako, da lahko vsako živilo kupiš 3x.</li> <li>Igralec naj ima na voljo več denarja.</li> <li>Na koncu dekle pove še, koliko katerih izdelkov si kupil. Npr. »Kupil/a si 2x lubenico, 1x grozdje, 2x krompirček«.</li> </ul>
Učni	Primer celotne aktivnosti v Snap!-u:
pripomočki,	https://snap.berkeley.edu/project?user=mateja&project=C4G16
sredstva za	<u>KupovanjeHraneZaPiknik</u>
učitelja	<ul> <li>Primer dodatne aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=mateja&amp;project=C4G16</u></li> </ul>
	KupovanjeHraneZaPiknik%20%2B%20Dodatek
	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana:</li> </ul>





	MK.
Učni	Navodila za učenca (C4G16_NavodilaZaUcenca.docx)
pripomočki	
za učenca	





# Učni scenarij 17 – Računanje

Naslov učnega	Računanje
scenarija	
Pričakovano	Uporaba spremenljivk za štetje točk in izbiranje videza ozadja in lika.
programersko	Uporaba naključnih števil za izbiro naključnega ozadja in videza lika.
predznanje	Uporaba ponavljajoče zanke.
	Uporaba pogojnega stavka.
	Uporaba primerjalnih operatorjev.
	Uporaba bloka za zaznavanje dialoga (vprašaj in čakaj).
	Uporaba bloka za pošiljanje dogodkov (pošlji vsem).
Učni cilji	Splošni učni cilji:
	• spremenljivke
	• pogojni stavki
	• zanke
	• posiljanje dogodkov
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>Učenci uporabijo spremenljivke za štetje točk.</li> <li>Učenci uporabijo spremenljivke za videz ozadja in lika</li> </ul>
	<ul> <li>Učenci uporabijo pogoje in logične operacije.</li> </ul>
	<ul> <li>Učenci uporabijo pošiljanje dogodkov za spremembo lika in izračun končnega rezultata.</li> </ul>
Cilji, naloge in kratek	Kratek opis: Med igro preverimo, ali igralec obvlada računske
opis aktivnosti	operacije v Snap!-u. Pravila so naslednja: igralec bo zaporedoma dobil
	za izračunati 10 naključnih računov, pri čemer bo prvo število vedno 6,
	naključno pa bo vsakič izbiralo med računskimi operacijami
	(seštevanje, odštevanje, množenje, deljenje) in drugim številom, ki pa
	je lahko 1, 2 ali 3. Igralec mora vnesti izračunani rezultat, pri čemer se
	mu pravilni in napačni odgovorijo štejejo. Po desetem izračunu, se
	izpiše koliko točk je igralec dosegel.





	Naloge: Učenci morajo določiti ozadje in videz lika, načrtovati
	potrebne spremenljivke in določiti katere bloke bodo potrebovali.
	Nato morajo sestaviti kodo ozadja in lika.
	Dodatna naloga: Učenci lahko na koncu igre dodajo povratno
	informacijo, kot je: »Odlično!« ali »Računskih operacij ne poznate še
	dobro!«.
	Cilji: Učenci bodo izboljšali svoje predhodno pridobljeno znanje z
	poznavanjem spremenljivk, naključnih števil, zank, pošiljanja
	dogodkov.
Trajanje aktivnosti	45 minut
Učne strategije in	Aktivni pouk (diskusija, eksperimentiranje z že pripravljeno igro),
metode	učenje z ustvarjanjem iger, problemsko učenje
Učne oblike	Individualno delo / delo v paru / frontalna oblika
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	
	MOTIVACIJA - UVOD
	1. Učitelj za uvodno motivacijo učencem poda v reševanje že
	ustvarjeno igro.
	(Peetro CTR)
	(mextre free
	6 / 2
	Zapiłi rezultat:
	2. Učitelj se z učenci pogovori o pogojih naloge. V igri je desetkrat
	število pa je izbrano naključno, od 1 do 3. Igralec vnese
	rezultat. Pravilni in napačni rezultati se štejejo. Na koncu igre
	igralec dobi povratno informacijo kako uspešen je bil pri računanju
	3. Učitelj se z učenci pogovori o spremenlijvkah, na kakšen način
	so definirane, nastavljene in kako se spreminjajo.

















Seštevanje:	
če costume name – of Ozadje – = • če odgovor = 6 + število spremeni sprementjivko Pravino × za 1 sicer spremeni sprementjivko Napačno × za 1	
Odštevanje:	
če costume name – of Ozadje – = -	
če odgovor = 6 – število spremeni spremenijivko Pravilno – za 1 sicer spremeni spremenijivko Napačno – za 1	
Množenje:	
če costume name v of Ozadje v = 1 če odgovor = 6 × število spremeni spremenijivko Pravilno v za 1 sicer spremeni spremenijivko Napačno v za 1	
Deljenje:	
če costume name v of Ozadje v = ) če odgovor = 6 / število spremeni spremenljivko Pravilno v za 1 sicer spremeni spremenljivko Napačno v za 1	





Za podajanje povratne informacije igralcu, kako je bil uspešen pri računanju, je potrebno najprej poslati sporočilo »seštej dosežene točke« vsem skriptam v orodju Snap!.

pošlji seštej dosežene točke - vsem

Ko lik *Število* sprejme sporočilo, sporoči igralcu koliko točje je dosegel. Dosežene točke izračuna tako, da število napačnih odgovorov odšteje od števila pravilnih odgovorov.

Skripta lika Število:



OPOMBA: V primeru, da imajo učenci primero predznanje o poznavanju računskih operacij, lahko nalogo nadgradimo z dodajanjem različnih naključnih iger, pri čemer je potrebno pri odštevanju in deljenju upoštevati dodatne pogoje (npr. zmanjševanec mora biti večje od odštevanca, števili morata biti deljivi). V nalogo lahko vključimo tudi operacijo modulo (mod).











	Lik Število
	ko sprejmem spremeni število - nastavi število - na naključno število od ) do 3 Preklopi na obleko število ko sprejmem seštej dosežene točke - reci poveži Doseglarsi (Pravilno - Napačno) točki + za 10 sekund.
Učni pripomočki, sredstva za učitelja	<ul> <li>Primer celotne aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=tadeja&amp;project=</u> <u>Racunanje</u></li> </ul>
	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>
Učni pripomočki za učenca	<ul> <li>Predloga aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=tadeja&amp;project=</u> <u>Racunanje_0</u></li> </ul>
	<ul> <li>Navodila za učenca (C4G17_NavodilaZaUcenca.docx)</li> </ul>





# Učni scenarij 18 – Recikliranje

Naslov učnega	Recikliranje
scenarija	
Pričakovano	Postavljanje lika na določeno mesto na odru
programersko	Prikazovanje in skrivanje lika
predznanje	Uporaba pogojnega stavka
	Uporaba preverjanja dotikanja
	Uporaba spremenljivk
	Uporaba zank
	Pošiljanje in prejemanje sporočil
Učni cilji	Splošni učni cilji:
	<ul> <li>Razdeli igro na manjše dele in z njimi sestavi celoto</li> <li>Podvoji in ustrezno prilagodi del kode</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>Učenec uporabi spremenljivko za štetje pravilno razvrščenih odpadkov in za preverjanje, ali je igralec že pospravil vse odpadke</li> <li>Učenec za vsak odpadek preveri, če se lik dotika drugega lika</li> </ul>
	<ul> <li>Učenec ustrezno skrije in pokaže lik</li> </ul>
	<ul> <li>Učenec nastavi ali je lik vlečljiv ali ne</li> </ul>
	<ul> <li>Učenec uporabi pošiljanje sporočil za interakcijo med liki</li> </ul>
	<ul> <li>Učenec uporabi zanko ponavljaj, dokler (ni odpadek v ustreznem košu)</li> </ul>
	<ul> <li>Učenec uporabi čakaj, dokler (niso pobrani vsi odpadki)</li> </ul>
Cilji, naloge in kratek	Cilj: Načrtovanje izdelave igrice
opis aktivnosti	
	Naloge: Za vsak lik napiši, kaj je njegova naloga. Označi, katere naloge
	že znaš sestaviti. Sestavi igro in jo testiraj.
	Kratek opis: Učencem predstavimo igrico, ki jo bodo izdelovali v tej uri
	in jim naročimo, naj razmislijo, kaj mora narediti posamezen lik.





	Skupaj pregledamo, kaj je potrebno narediti. Učenci označijo kaj že
	znajo in prosijo za pomoč, če se jim kje zatakne. Sestavijo igrico po
	svoji predlogi in preverijo pravilnost njenega delovanja.
Trajanje aktivnosti	45 minut
Učne strategije in	Možganska nevihta
metode	Razgovor
	Metoda praktičnega dela
Učne oblike	Frontalno delo
	Individualno delo
	Delo v paru
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	[predstavitev igre]
	Robot Smetko je opazil, da na otroškem igrišču ležijo papirnati in
	stekleni odpadki in se zato otroci tam ne morejo brezskrbno igrati. Ker
	želi otrokom in njihovim staršem pomagati pri čiščenju igrišča, je
	prinesel dve posodi, v kateri je potrebno razvrstiti odpadke z igrišča - v
	zeleni koš sodijo odpadki iz stekla, v modrega pa odpadni papir.
	Sestavi igrico, s katero se bodo otroci naučili, v kateri zabojnik morajo
	odvreči posamezen odpadek, da bo igrišče spet čisto, odpadki pa
	pospravljeni v ustrezno posodo.
	[Načrtovanje igre]
	Za izdelavo igre boš uporabil, a naslednje like: robota Smetka, dva
	koša za odnadke in dve vrsti različnih odnadkov. Bazmisli, kaj je naloga
	posameznih likov (kaj morajo parediti, kdaj morajo biti vidni, kdaj ali
	jih smemo vleči ) in to zaniši na učni list
	Skunaj nogledamo, kaj so nanisali in no notrehi donolnimo, nonravimo
	Primor ročitvo:
	Robot Smotko
	- Boyo zgodbo in povodilo zo igro
	<ul> <li>Pove zgodbo in navodna za igro</li> <li>Sporoči, da se igra začenja</li> <li>Šteje pospravljene smeti</li> </ul>





	<ul> <li>Počaka na konec igre</li> <li>Pove rezultat</li> <li>Prikazan na začetku in koncu, skrit med igro</li> </ul>
	Kosa za smeti
	<ul> <li>Se prikažeta na določenem mestu</li> <li>Ne moremo jih vleči</li> </ul>
	Smeti
	<ul> <li>Pokaže se na določenem mestu</li> <li>Lahko jo vlečemo</li> <li>Preverja, če je v napačnem košu in opozori, če je</li> <li>Preverja, če je v pravem košu, in ko je se skrije in šteje za pospravljeno</li> </ul>
	Med naštetim nalogami likov označite tiste, za katere menite, da jih
	znate narediti sami. O ostalih razmislite, kaj vas zanima in si na učni
	list zapišite vprašanja, ki jih imate. O teh vprašanjih se pogovorite s
	sošolcem. Če tudi on_a ne zna odgovoriti, se posvetujeta z učiteljem.
	[Izdelava igre]
	V Snap!-u sestavite igro, kot ste jo načrtovali. Pri izdelavi si lahko
	pomagate s predlogo
	(https://snap.berkeley.edu/snap/snap.html#present:Username=spela
	<u>c&amp;ProjectName=recikliranje_predloga</u> ), v kateri lahko najdete vse like.
	Posamezne dele igre sproti testirajte, da vidite, če delujejo pravilno.
	[Refleksija]
	Vprašamo učence, kaj jim je bilo pri takšni izdelavi igre všeč, kaj so
	pogrešali, če so imeli kakšne težave, kako so jih rešili. Kako bi igro
	nadgradili?
Učni pripomočki,	<ul> <li>Primer končne rešitve osnovnega primera v Snap!-u:</li> </ul>
sredstva za učitelja	<u>https://shap.berkeley.edu/shap/shap.html#present:</u> Username=spelac&ProjectName=C4G_18_recikliranje
Učni pripomočki za	<ul> <li>Predloga aktivnosti v Snap!-u:</li> </ul>
učenca	https://snap.berkeley.edu/snap/snap.html#present:
	<ul> <li><u>Username=spelac&amp;ProjectName=recikliranje_predloga</u></li> <li>Učni list za učenca (C4G18_Ucnil istZaUcenca docx)</li> </ul>
	<ul> <li>Navodila za ucenca, ki je malo manj vesc programiranja iger (C4G18_NavodilaZaUcenca.docx)</li> </ul>





## Učni scenarij 19.1 – Zaigraj na klavir 1

Naslov učnega	Zaigraj na klavir 1
scenarija	
Pričakovano	Uporaba spremenljivk za štetje točk
programersko	Uporaba dogodka <i>ko je gumb miška pritisnjen</i>
predznanje	Uporaba zanke
	Uporaba pogojnega stavka
	Uporaba pošiljanja obvestil za menjavo ozadij in upravljanje likov
Učni cilji	<ul> <li>Splošni učni cilji:</li> <li>Spremenljivke</li> <li>Pogojni stavek</li> <li>Zanke</li> <li>Pošiljanje obvestil</li> <li>Zvok</li> <li>Sestavljanje kode za melodijo</li> <li>Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:</li> <li>Učenec uporabi spremenljivko za štetje točk.</li> <li>Učenec uporabi ukaz za pošiljanje obvestil za menjavo ozaij in upravljanje likov.</li> <li>Učenec uporabi ukaze za zvok za ustvarjanje melodije.</li> <li>Učenec uporabi zanko za zmanjšanje števila ukazov v kodi.</li> <li>Učenec razširi funkcionalnost igre.</li> </ul>
Cilji, naloge in	Kratek opis: Pojdimo v čudoviti svet kraljice Mary. Igralca povabi v grad, kjer
kratek opis	bosta poslušala glasbo. V plesni dvorani njen prijatelj Dino igra na klavir.
aktivnosti	Dino zaigra nekaj tonov, igralec pa mora ugotoviti, kateri ton je Dino zaigral.
	Pravilen odgovor prinese 1 točko. Nato Dino zaigra pesem, igralec pa mora
	ugotoviti njen naslov. Pravilen odgovor prinese 5 točk. Naloge: Učenci imajo na voljo predlogo z ozadji in liki (z več oblekami).
	Ustvariti morajo spremenljivko za štetje točk, ugotoviti, katere ukazne bloke
	potrebujejo, se spoznati z ukaznimi bloki za zvok ter napisati kodo za
	predvajanje melodij/e.
	Cilji: Učenci se bodo naučili napisati kodo za ustvarjanje melodije ter




	izboljšali svoje znanje o spremenljivkah, zankaj, pogojnih stavkih in
	dogodkih.
Trajanje	90 minut
aktivnosti	
Učne strategije	Aktivno učenje (diskusija, eksperimentiranje z vnaprej pripravljeno igro),
in metode	učenje z izdelavo iger, reševanje problemov
Učne oblike	Individualna učna oblika / Delo v paru / Skupinska učna oblika
	Frontalna učna oblika s sodelovanjem celotnega razreda
Povzetek	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
učnega procesa	
	[Korak 1]
	Podamo uvodna navodila: Naredili bomo igro, kjer
	se kraljica Mary pojavi pred gradom in igralca
	povabi v grad. Po prihodu v plesno dvorano,
	dinozaver Dino na klavir zaigra nekaj tonov, ki jih
	mora igralec uganiti. Pravilen odgovor prinese 1 točko. Nato igralec klikne
	na klavir in Dino zaigra pesem, ki jo mora igralec prav tako uganiti. Sedaj
	pravilen odgovor prinese 5 točk. Kraljica na koncu pove, koliko točk je
	igralec dosegel. Če igralec ugane pesem, Dino zapleše.
	[Korak 2]
	Predvajamo delen program:
	https://snap.berkeley.edu/project?user=mateja&project=C4G19.1 ZaigrajN
	<u>aKlavir 1</u>
	Pogovorimo se, kaj v igri še manjka, glede na začetna navodila (odhod v
	grad, Dinov ples, Mary pove število doseženih točk). Kako bomo napisali
	kodo za predvajanje glasbe? Ker tega še ne znamo, si bomo skupaj pogledali
	v naslednjem koraku.
	[Korak 3]
	Učencem razložimo, kako delujejo ukazi za zvok. Uporabimo ukaz





play note 60 v for 0.5 beats

Številka 60 pomeni, kateri ton bo predvajan, 0.5 beats pa pomeni, koliko časa bo ton predvajan. Ni potrebno, da poznamo številke (60) tonov na pamet, saj se nam s klikom na puščico poleg številke pokaže klavir, kjer izberemo ustrezen ton.



V desnem polju določamo dolžino tona. Vpišemo lahko (decimalno) število, ali pa število v obliki ulomka.

play note 60 for 0.5 beats ali

Učenci lahko sami poskušajo in določajo trajanje tona.

### [Korak 4]

Učenci bodo napisali kodo za ugibanje zaigranega tona. Ton se bo najprej zaigral, nato se pojavi vprašanje. Če je odgovor pravilen, prištejemo točko. Dodamo tudi povratno informacijo o pravilnem / nepravilnem odgovoru. Koda lahko izgleda tako:



Korak lahko večkrat ponovimo, pri tem zamenjamo zaigran ton in odgovor. Celotna koda se lahko izvede s prejemom obvestila, npr.











alay note 55 p for 1 / A best





1



Sofinancira program Evropske unije Erasmus+

ko sprejmem klaviri – play note (48 – for (1) beats vprašaj Katero noto sem zaigral, *CaliF2 in čakaj	
cs odgovor   reci Pravinoi   spremeni spremenijivka   sicer   reci Napačnoi   za 2   sekund.   play note   52   for   1   sicer   reci   Napačnoii   za   2   sekund.         (a) odgovor (a) odgovor (b) odgovor (b) odgovor (c) odgo	
vprašaj Katero noto-sem zaigral? in čakaj če odgovor = G reci Pravinoj za 2 sekund. spremeni spremenijivko tocke za 1 sicer reci Napačnoji za 2 sekund. reci Sodaj paruganinaslov posmi. Klikni na klavić za 6 sekund.	
Ozadje	





	ko kilknemo na Preklopi na obleko grad ko sprejmem Pojdi v plesno dvorano Preklopi na obleko plesna_dvorana
Učni pripomočki, sredstva za učitelja	<ul> <li>Primer celotne aktivnosti v Snap!-u: https://snap.berkeley.edu/project?user=mateja&amp;project=C4G19.1 Z</li> <li>aigrajNaKlavir</li> <li>Delna aktivnost v Snap!-u: https://snap.berkeley.edu/project?user=mateja&amp;project=C4G19.1 Z</li> <li>aigrajNaKlavir 1</li> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>
Učni pripomočki za učenca	<ul> <li>Predloga aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=mateja&amp;project=C4G19.1_Z</u> <u>aigrajNaKlavir_Delno</u></li> <li>Učni list za učenca (C4G19.1_UcniListZaUcenca.docx)</li> <li>Navodila za učenca (C4G19.1_NavodilaZaUcenca.docx)</li> </ul>





# Učni scenarij 19.2 – Zaigraj na klavir 2

Naslov učnega	Zaigraj na klavir 2
scenarija	
Pričakovano	Uporaba zanke
programersko	Uporaba spremenljivk
predznanje	Uporaba pogojnega stavka
Učni cilji	Splošni učni cilji: • Pogojni stavek • zanke
	<ul> <li>Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:</li> <li>Učenec zna uporabiti zanko za igranje glasbe</li> <li>Učenec zna napisati kodo, da ob pritisku tipke program predvaja zvok</li> <li>Učenec doda zvok liku</li> <li>Učenec zna zamenjati obleko lika s pritiskom na tipko</li> </ul>
Cilji, naloge in	Kratek opis: Učenec po danih notah zaigra pesem na klavir.
kratek opis	Naloge: Učenec napiše kodo za klavirske tipke – vsaka tipka (lik) mora
aktivnosti	zaigrati določen ton. Na zaslonu so še trije liki: violinski ključ, ki pokaže
	besedilo z notami; nota, ki zaigra pesem; X, ki izbriše besedilo pesmi.
	Cilji: Učenci se bodo naučili, kako se predvaja glasba ter kako se zamenja
	obleka liku s klikom nanj.
Trajanje	45 min
aktivnosti	
Učne strategije in	Aktivno učenje, učenje z izdelavo iger, reševanje problemov
metode	
Učne oblike	Individualna učna oblika / Delo v paru
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	
	Učencem na začetku pokažemo, kako bo izgledala končna igra.
	Ko odpremo projekt, imamo na ekranu klavir, violinski ključ, noto ter gumb
	X (slika levo). S klikom na violinski ključ se pokaže besedilo z notami (slika





desno), s klikom na noto se zaigra celotna pesem, s klikom na gumb X pa se ozadje z besedilo zamenja z belim ozadjem, tako da imamo ponovno sliko na levi.

S klikom na posamezno belo tipko se zaigra določen ton in tipki se za delček sekunde spremeni obleka.





TWINKLE TWINKLE LIFTLE STAR HOW I WONDER WHAT YOU ARE

Učencem je na začetku dano:

- Tipka C, ki je na svojem mestu in ima obleke za vse manjkajoče tipke
- Črna tipka
- Belo ozadje in ozadje z akordi
- Vsi trije liki: violinski ključ, nota ter gumb X

### [Korak 1]

Odpremo predlogo in sestavimo klavir. Tipko C podvojimo in povlečemo na svoje mesto. Zamenjamo ji obleko. Nato podvojimo še črno tipko in jo postavimo na svoje mesto. Postopek ponavljamo, dokler nimamo vrhnje (leve) slike. Za lažje delo v nadaljevanju, like preimenujemo.































	<ul> <li>Učenec doda dodatne naloge po svojih željah ali sledi spodnjim navodilom:</li> <li>Podvoji lik Nota (ter zamenjaj položaj lika na ozadju) ter napiši program, ki bo zaigral drugo pesem.</li> <li>Dodaj ozadje z besedilom in akordi za novo pesem.</li> </ul>
Učni pripomočki, sredstva za učitelja	<ul> <li>Primer celotne aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=mateja&amp;project=C4G19.2</u></li> <li><u>ZaigrajNaKlavir%20-%20Cela</u></li> <li>Lajovic, S. (2011). Scratch. <i>Nauči se programirati in postani</i> <i>računalniški maček</i>. Ljubljana: Pasadena.</li> <li>Vorderman, C. (2017). <i>Računalniško programiranje za otroke</i>. Ljubljana: MK.</li> </ul>
Učni pripomočki za učenca	<ul> <li>Predloga aktivnosti v Snap!-u: <u>https://snap.berkeley.edu/project?user=mateja&amp;project=C4G19.2</u></li> <li><u>ZaigrajNaKlavir%20-%20Delna</u></li> <li>Navodila za učenca (C4G19.2_NavodilaZaUcenca.docx)</li> </ul>





# Učni scenarij 20 – Test

Naslov učnega	Test
scenarija	
Pričakovano	Postavljanje lika na določeno mesto na odru
programersko	Prikazovanje in skrivanje lika
predznanje	Uporaba pogojnega stavka
	Uporaba spremenljivk
	Pošiljanje in prejemanje sporočil
Učni cilji	Splošni učni cilji:
	<ul> <li>Razdeli igro na manjše dele in z njimi sestavi celoto</li> <li>Zamenjaj ozadje</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>Učenec uporabi delček vprašaj, da dobi odgovor na vprašanje</li> </ul>
	<ul> <li>Učenec uporabi pogojni stavek za preverjanje pravilnost odgovora</li> </ul>
	<ul> <li>Učenec uporabi spremenljivki za štetje pravilnih in napačnih odgovorov</li> <li>Učenec uporabi pošiljanje sporočil za menjavanje ozadja</li> </ul>
	<ul> <li>Učenec uporabi delček za računanje razlike</li> </ul>
Cilji, naloge in kratek	Cilj: Sestaviti preverjanje znanja za sošolca
opis aktivnosti	Naloge: Sestavi test, s katerim boš preveril sošolčevo znanje o Snap!-u
	Kratek opis: Učencem predstavimo nalogo: sestavite test z vprašanji,
	ki so prikazana kot ozadja na odru.
Trajanje aktivnosti	45 minut
Učne strategije in	Metoda praktičnega dela
metode	
Učne oblike	Frontalno delo
	Individualno delo





Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	[predstavitev naloge]
	Abby bi rada preverila sošolčevo znanje o Snap!-u. Sestavila je test z
	vprašanji, koda za zastavljanje vprašanj in preverjanje pravilnosti
	odgovorov pa se ji je izbrisala. Vprašanja je shranila kot različna
	ozadja. Pomagaj ji sestaviti program, v katerem se bo za vsakim
	vprašanjem pokazalo naslednje, lik Abby pa bo sošolcu povedal, s
	katerimi besedami lahko odgovori. Na koncu naj program sošolcu
	pove, kolikokrat je odgovoril pravilno in kolikokrat napačno, ter število
	točk, ki jih je dosegel (vsak pravilen odgovor prinese 1 točko, vsak
	napačen pa -1). Abbyjin program najdeš tukaj:
	https://snap.berkeley.edu/snap/snap.html#present:
	Username=spelac&ProjectName=C4G 20 test tmp
	[Načrtovanje]
	Iz opisa si skicirajte, kako mora izgledati program. Lahko si naloge
	zapišete po alinejah ali na način, ki je vam bližji. Razmislite, kako bo
	oder izvedel, kdaj naj zamenja ozadje.
	Primer načrta:
	Oder:
	- Na začetku pokaže začetno ozadje
	- Ko prejme sporočilo, zamenjavo ozadja na naslednje
	Gump:
	<ul> <li>Se na začetku prikaze</li> <li>Ob kliku nanj pošlje sporočilo o začetku in se skrije</li> </ul>
	Abby:
	- Nastavi vse točke na 0
	- Pove navodila
	<ul> <li>Ob kliku na gumb začne s postavljanjem vprašanj:</li> <li>Sporoči odru, poj zamonja ozadjo za poslodnje vprašanjo</li> </ul>
	<ul> <li>Sporoci ouru, naj zamenja ozacije za naslednje vprasanje</li> <li>Postavi vprašanie z možnimi odgovori</li> </ul>
	<ul> <li>Preveri ali je odgovor pravilen</li> </ul>
	<ul> <li>Če je, prišteje točko k pravilnim odgovorom</li> </ul>





	<ul> <li>Če ne, prištej točko k napačnim odgovorom</li> <li>Pred novim vprašanjem zamenja ozadje, pove vprašanje, preveri odgovor – to ponovi večkrat</li> <li>Na koncu pove število pravilnih odgovorov, število napačnih odgovorov, število točk</li> </ul>
	V Shap!-u sestavite preverjanje znanja kot ste ga nacrtovali. Pri
	izdelavi si lahko pomagate s predlogo
	(https://snap.berkeley.edu/snap/snap.html#present:
	<u>Username=spelac&amp;ProjectName=C4G_20_test_tmp</u> ), v kateri lahko
	najdete vse like in začetna Abbyjina navodila. Posamezne dele
	programa sproti testirajte, da vidite, če delujejo pravilno.
	[Dodatne naloge – če je čas]
	Dodatna naloga 1: Med igro Abby večkrat spremeni svoj izgled.
	Dodatna naloga 2: Sami sestavite podobno preverjanje znanja. Ker v
	Snap!-u ni možnosti za pisanje besedila, lahko slike za ozadje
	pripravite v poljubnem programu za izdelavo rastrske (npr. Slikar,
	Paint.net) ali vektorske grafike (npr. Inkscape) in jih uvozite v svoj kviz
	Snap!-u. Če slike niso potrebne, lahko vprašanje v celoti zastavi kar
	Abby.
	[Refleksija]
	Vprašamo učence, kaj jim je bilo pri takšni izdelavi igre všeč, kaj so
	pogrešali, če so imeli kakšne težave, kako so jih režili. Kako hi igro
	nadgradili?
Llčni prinomočki	Primor končno ročitvo osnovnoga primora v Snanl. u:
	• Frinter Konche resitve osnovnega primera v snapi-d. https://snap.berkeley.edu/snap/snap.html#present:
sredstva za ucitelja	Username=spelac&ProjectName=C4G 20 test
Učni pripomočki za	<ul> <li>Predloga aktivnosti v Snap!-u:</li> </ul>
 Učenca	https://snap.berkeley.edu/snap/snap.html#present:
accilia	Username=spelac&ProjectName=C4G_20_test_tmp
	Ocni list za ucenca (C4G20_OcniListZaUcenca.docx)
	<ul> <li>Navodila za učenca, ki je malo manj vešč programiranja iger (C4G20_NavodilaZaUcenca.docx)</li> </ul>





## Učni scenarij 21 – Enostavni PACMAN

Naslov učnega	Enostavni PACMAN
scenarija	
Pričakovano	<ul> <li>pogojni stavki,</li> </ul>
programersko	<ul> <li>programiranje več objektov,</li> </ul>
predznanje	• zaznavanje barve,
	<ul> <li>zanke (neskončna, ponavljaj dokler)</li> </ul>
	<ul> <li>premikanje z uporabo dogodkov,</li> </ul>
	<ul> <li>naključna števila</li> </ul>
Učni cilji	Splošni učni cilji
	<ul> <li>kloniranje objektov,</li> </ul>
	<ul> <li>definiranje obnašanja klona,</li> </ul>
	<ul> <li>pošiljanje sporočil med objekti,</li> </ul>
	<ul> <li>boolean vrednosti v logičnih izrazih,</li> </ul>
	<ul> <li>definiranje, razlikovanje, dinamično preverjanje in odziv na dve različni stanji igre.</li> </ul>
	Specifični učni cilji, ki so osredotočeni na algoritmično mišljenje:
	<ul> <li>učenec implementira premikanje objekta z uporabo smernih tipk in dogodkov, pri čemer upošteva omejitve,</li> </ul>
	<ul> <li>učenec uporabi kloniranje, da naredi klone nekega objekta,</li> </ul>
	<ul> <li>učenec zna določiti obnašanje klonov,</li> </ul>
	<ul> <li>učenec pozna pomen pošiljanja sporočil med objekti,</li> </ul>
	<ul> <li>učenec zna povezati pošiljanje sporočil iz klona za to, da poveča števec, ki je v nekem drugem objektu,</li> </ul>
	<ul> <li>učenec zna ugotoviti, da je objekt prejel sporočilo in se takrat ustrezno odzvati.</li> </ul>
Cilji, naloge in kratek	Kratek opis: Sprogramirajte igro, v kateri bo igralec pobiral zvezde, ki
opis aktivnosti	se bodo naključno pojavljale v labirintu in se pri tem izogibal duhcu.
	Naloge: Učenci morajo implementirati premikanje glavnega karakterja
	v igri, tako da se bo premikal zgolj znotraj labirinta in ne bo mogel iti
	čez stene. Nato morajo ustvariti objekt zvezdo, na začetku igre





ustvariti njen klon na naključni lokaciji, pri čemer morajo paziti, da se ustvari na dovoljeni lokaciji, nato pa to ponoviti vsakič, ko igralec zvezdo pobere. Prav tako morajo beležiti število pobranih zvezd in končati z igro, ko jih igralec pobere dvajset. Igra bo bolj zanimiva, če vanjo dodajo še duhca, ki se bo po labirintu naključno premikal. V primeru, da se bo glavni karakter tega duhca dotaknil, pa bo igre takoj konec.

S to aktivnostjo bodo učenci ponovili znanje o premikanju objekta znotraj labirinta, kar so se naučili v eni od prejšnjih aktivnosti. Spoznali bodo koncept kloniranja objekta z upoštevanjem omejitev pri naključnem pozicioniranju na ekranu. Naučili se bodo narediti računalniško voden lik, ki se bo naključno premikal.

Trajanje aktivnosti	90 minut
Učne strategije in	active learning, collaborative learning, problem solving
metode	
Učne oblike	frontalni način podajanja snovi
	individualno delo/delo v parih/skupinsko delo
Povzetek učnega	(Motivacija-uvod, Implementacija, Refleksija in vrednotenje)
procesa	Igralec zbira zvezdice, ki se naključno pojavljajo na ekranu, ob tem pa
	ga lovi rdeč duhec. Če glavni lik - Pacman trči z duhcem je igre konec
	(neuspešno), drugače pa je igre konec, ko zbere 20 točk (uspešno).
	[1. korak]
	Učenci izdelajo labirint v katerem je
	območje, po katerem se lahko premika
	glavni lik, v celoti neke barve (npr. modre),
	zidovi, ki ga ustavijo, pa so neke druge barve
	(npr. črne). Če želimo prihranit s časom, jim
	slikovni material pripravimo vnaprej.





#### [2. korak]

Nato morajo narisati glavni lik - Pacman-a in pa rdečega duhca. Zvezdo lahko predstavlja majhen krogec, ki ga lahko narišejo znotraj programa Snap!.



### [3. korak]

Za programiranje premikanja glavnega lika lahko uporabimo več različnih možnosti. Spodnji primer prikazuje eno od njih. V njem uporabimo dogodke za ugotavljanje ali je igralec pritisnil določeno tipko, nato preverimo, kje se nahaja glavni lik. Če se dotika ustrezne barve potem ga obrnemo v smer premika in opravimo korak. Če se po koraku dotakne nedovoljene barve, ki označuje zid, pa ga premaknemo za enako dolg korak v nasprotno smer.







[4. korak]

Zdaj pa se lotimo programiranja zvezd. Vsaka od zvezd bo imela enak izgled in obnašanje. To je tipičen primer v katerem uporabimo klone, saj bi bilo kopiranje istega objekta neučinkovito in nerodno. V našem primeru bi morali to narediti dvajsetkrat, če bi želeli igro podaljšati na sto pobranih zvezdic, pa bi bilo to na ta način že skoraj nemogoče. Na začetku igre se bo ustvaril nov klon, ki se bo pojavil naključno na neki dovoljeni lokaciji v labirintu. Ko ga bo igralec pobral, bo izginil, nato pa se bo pojavil nov klon na novi lokaciji. Prvi klon na začetku igre bomo naredili z uporabo spodnje kode, ki jo bomo dali na ozadje:



Ker ne želimo, da bi se originalni objekt zvezde prikazal, ampak želimo to le za klone, ga na začetku igre skrijemo. Nato se lotimo iskanja ustrezne lokacije na katero lahko postavimo zvezdo v labirintu. Če bi namreč zvezdo npr. postavili na zid, igralec do nje sploh ne bi mogel. Strategija, da dosežemo to je naslednja:

- 1. Pozicija objekta na zaslonu je izražena preko x in y koordinate, ki lahko imata katerokoli vrednost na intervalu [-140, 140], če dimenzij zaslona ne spreminjamo oz. ustrezno spremenjenem, če se za to odločimo.
- 2. Nato preverimo, če se klon dotika barve zidu.
- Če se ne dotika barve zidu ga prikažemo (spomnimo, da smo original na začetku skrili, kar pomeni, da so tudi kloni privzeto skriti.) in v neskončni zanki preverjamo, če se je dotaknil glavnega lika.
- Če se dotika barve zidu, ta lokacija ni dobra, zato naredimo nov klon (v upanju, da bo tokrat na boljši lokaciji), tega pa izbrišemo. To se bo ponavljalo dokler si ne bo izbral dovoljene lokacije.
- 5. Ko igralec zvezdo pobere, moramo števec zvezd povečati za ena. Ta števec mora biti v nekem objektu, ki ni klon, saj ga ne







4. 270 degrees - LEVO





Opazimo, da so ta števila večkratniki števila 90, zato lahko naključnega med njimi dobimo tako, da si najprej izmislimo naključno število od 0 do 3, nato pa ga pomnožimo z 90. Vrednost, ki smo jo na ta način dobili predstavlja novo naključno smer premikanja.

Duhec se premika dokler ne trči ob glavni lik. To lahko uporabimo kot pogoj v zanki - *ponavljaj dokler*. Ko se to zgodi, pa je igre konec.

ko kliknemo na 🍋
nastavi smer na 🛛
ponavljaj, dokler (se dotika pacman v ?)
obrni se v smeri smer premakni se 1 korakov če se dotika 2
premakni se 🔄 korakov
nastavi smer 🔻 na 🛛 naključno število od 🕕 do ₃ 🗙 90
reci Nitiruspelo, igreje konec. za 2 sekund.
ustavi vse 💌

### [6. korak]

Na koncu še poskrbimo za prištevanje točk in ugotavljanje kdaj je igralec uspešno končal igro. Števec smo implementirali znotraj kode glavnega lika. Na začetku smo ga postavili na 0, nato pa mu za vsakič, ko je klon oddal sporočilo, da se je dotaknil glavnega lika, povečali za 1. Ko se števec poveča samo preverimo, če je že dosegel vrednost 20. Takrat je igre konec.







Orodja in viri za učitelje	<ul> <li>Celotna aktivnost v programu Snap!: <u>https://snap.berkeley.edu/project?user=zapusek&amp;</u> project=enostavni pacman</li> </ul>
	<ul> <li>Lajovic, S. (2011). Scratch. Nauči se programirati in postani računalniški maček. Ljubljana: Pasadena.</li> </ul>
	<ul> <li>Vorderman, C. (2017). Računalniško programiranje za otroke. Ljubljana: MK.</li> </ul>
Viri/gradiva za učence	<ul> <li>Predloga aktivnosti v Snap!-u: https://snap.berkeley.edu/project?user=zapusek&amp; project=pacman_template</li> </ul>





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.