

RAZVIJANJE RAČUNALNIŠKEGA MIŠLJENJA SKOZI GLASBENE DEJAVNOSTI V OSNOVNI ŠOLI

¹Mojca Krevel

¹Osnovna šola Hruševce Šentjur, Slovenija

Povzetek

Ključne besede:

Računalniško mišljenje, programiranje, Scratch, glasbena umetnost, ustvarjanje, osnovna šola

Keywords:

computational thinking, programming, Scratch, music education, creativity, primary school.

Po mnenju mnogih strokovnjakov računalniško mišljenje predstavlja eno ključnih veščin 21. stoletja. Ta veščina določa način razmišljanja, ki omogoča strateško načrtovanje, kreiranje novih idej in učinkovito reševanje problemov. V šoli ga pri učencih razvijamo z vključevanjem temeljnih vsebin RIN pri pouku. Računalniško mišljenje se lahko učinkovito razvija tudi skozi glasbene dejavnosti, saj pristop glasbenega komponiranja vključuje algoritmično in abstraktno mišljenje, ki sta sestavni del računalniškega mišljenja. V prispevku opišemo, kako učinkovito je učenje tonske abecede, ritmičnih vrednosti in glasbenih simbolov s pomočjo programiranja. S študijo primera z opazovanjem in analizo izdelkov smo ugotovili, da je programiranje učinkovita metoda za učenje in utrjevanje glasbeno teoretičnega znanja in da učence močno motivira za aktivno sodelovanje pri pouku. Ugotovitve potrjujejo tudi rezultati ankete, v kateri učenci izrazijo navdušenje za programiranje in pozitivno ovrednotijo svoje rezultate. S prispevkom želimo spodbuditi predvsem učitelje glasbene umetnosti, da v svoj pouk vključijo aktivnosti, ki spodbujajo računalniško mišljenje.

DEVELOPING COMPUTATIONAL THINKING THROUGH MUSICAL ACTIVITIES IN PRIMARY SCHOOL

Abstract

According to many experts, computational thinking is one of the key competencies of the 21st century. It defines a way of thinking that enables strategic planning, the creation of new ideas, and effective problem-solving. In schools, pupils develop them by integrating fundamental computer science content into lessons. Computational thinking can be effectively developed through musical activities, as musical composition involves algorithmic and abstract thinking, which are essential components of computational thinking. This article

Copyright: © 2025

Avtorji/The author(s). To delo je objavljeno pod licenco Creative Commons CC BY Priznanje avtorstva 4.0 Mednarodna.

Uporabnikom je dovoljeno tako nekomercialno kot tudi komercialno reproduciranje, distribuiranje, dajanje v najem, javna priobčitev in predelava avtorskega dela, pod pogojem, da navedejo avtorja izvirnega dela.

<https://creativecommons.org/licenses/by/4.0/>



describes the efficiency of learning musical notation, rhythmic values, and musical symbols through programming. With a case study based on observation and analysis of pupils' work, we discovered that programming is an effective method for learning and reinforcing music theory knowledge, and it strongly motivates pupils to participate in lessons actively. The findings are supported by survey results in which pupils expressed enthusiasm for programming and positively evaluated their achievements. The paper aims to encourage music educators to incorporate activities that promote computational thinking into their teaching.

1 UVOD

Učni načrt za glasbeno umetnost med tematskimi sklopi za drugo vzgojnoizobraževalno obdobje med drugim izpostavlja notno opismenjevanje s ciljem, da učenci razumejo in uporabljajo glasbene pojme vezane na notno opismenjevanje. Učenci v šestem razredu naj bi tako znali uporabljati solmizacijo in tonsko abecedo in se orientirati v notnem zapisu. Učenci bi tako morali v šestem razredu razumeti in znati uporabljati glasbene simbole in poznati osnovne ritmične vrednote in tonsko abecedo (Holcar idr., 2011). V osnovni šoli so teoretične vsebine učencem precej abstraktne. Večina učencev ne uspe ponotranjiti vseh zakonitosti glasbenega jezika, teorijo težko učinkovito povežejo z glasbeno izkušnjo.

Zato so precej nemotivirani za glasbeno ustvarjanje in poustvarjanje z notnim zapisom. V raziskavi iz leta 2016 (Stefanija idr., 2016) učitelji glasbe izpostavljajo tudi problem preobsežnosti teoretičnih vsebin, ki predstavljajo pretežni del glasbenih vsebin in tako ostaja premalo časa za praktični del; ustvarjanje, petje in poustvarjanje. Tako se že dalj časa kaže močna potreba za oblikovanje drugačnih, inovativnih pristopov učenja glasbe, ki bi učencem omogočili pridobivanje glasbenih znanj skozi ustvarjalno in aktivno udejstvovanje, pri čemer bi lahko povezali glasbene vsebine z učencevimi izkušnjami in potrebami. Učitelji v svoje delo ves čas vpeljujejo novosti in preizkušamo nove metode dela v razredu. V zadnjih letih so le-te pogosto povezane s pametnimi napravami, ki omogočajo učencem drugačen, bolj interaktiven pogled na učne vsebine. V tem šolskem letu smo tako prvič v glasbeni pouk vključili programiranje s programskim jezikom Scratch. Odkrivanje novega področja nas je pripeljalo do računalniškega mišljenja, ki ima veliko vzporednic z glasbenim komponiranjem.

2 KAJ JE RAČUNALNIŠKO MIŠLJENJE?

Računalniško mišljenje kot ključno veščino 21. stoletja postavljajo ob bok osnovnim učnim spretnostim, kot so branje, pisanje in računanje. Prav tako opredeljujejo, da je programiranje enako pomembna veščina kot pisanje, saj omogoča ljudem ustvarjati nove zgodbe, igre, animacije itd. Računalniško mišljenje je pomembno orodje ustvarjalnega in kritičnega mišljenja, odločanja ter reševanja problemov (Krajnc idr., 2017). Z uvajanjem računalniškega mišljenja v šolah omogočamo učencem, da na drugačen način spoznavajo tehnologijo, ki jih obdaja, in se jo naučijo vključevati v svoje rešitve. Tako se sistematično spreminjajo ne le v uporabnike, ampak tudi v ustvarjalce digitalnih orodij in naprav. S tem pa tudi urijo postopkovni oz. algoritemski način mišljenja. Računalniško mišljenje spodbuja tudi razvoj samoregulacijskega učenja oz. učenje učenja, ki je ena ključnih kompetenc vseživljenjskega učenja. Razvija samostojnost organizacije in usmerjanje lastnega učenja ter učinkovito upravljanje s časom, viri in podatki. Konsistentno urjenje v računalniškem mišljenju pomeni tudi dolgoročno opolnomočenje učencev za vztrajanje ob neuspehu in učinkovito strategijo spoprijemanja z negativnimi čustvi, saj razvijejo strateški način mišljenja, ki jim omogoči učinkovito reševanje problemov na vseh področjih svojega življenja. Neuspeh razumejo kot del procesa in rasti, zato lažje najdejo nove proložitosti in izzive ter nenehno stremijo k doseganju novih ciljev. Takšni učenci so kasneje poklicno in zasebno uspešnejši. Tisti učenci, ki pa ne razvijejo računalniškega mišljenja, so v svojih sposobnostih reševanja problemov zelo omejeni. Glede na številne pozitivne učinke je smiselno, da nudimo učencem čim več aktivnosti, ki spodbujajo razvoj računalniškega mišljenja (Krajnc idr., 2017).

Različni avtorji navajajo različne osnovne veščine računalniškega mišljenja. Selby in Woollard (2013, v Koron, 2021) vključita algoritmično razmišljanje, logično razmišljanje, dekompozicijo, abstrakcijo in posploševanje.

2.1 KAKO RAZVIJAMO RAČUNALNIŠKO MIŠLJENJE V ŠOLI?

V šoli računalniško mišljenje razvijamo s programiranjem v programih, kot so Scratch, LEGO education SPIKE ali Python. Računalniško mišljenje lahko razvijamo tudi s pomočjo izobraževalnih robotkov kot so Kubo, Cubetto in mTiny. Lahko pa računalniško mišljenje spodbujamo tudi z aktivnostmi brez uporabe računalnika. Ena takšnih metod je Računalništvo brez računalnika (ang. Computer Science Unplugged), kjer učenci pridobivajo temeljna znanja RIN skozi gibanje in uporabo različnih tehnik. Druga metoda pa je reševanje logičnih nalog, s katerimi se učenci na osnovni šoli najpogosteje srečajo v

okviru tekmovanja Bober (skupina RINOS, b. d.). Pri pouku glasbe smo uporabljali Scratch, ki je vizualni programski jezik, enostaven za učenje programiranja. V njem učencem ni treba pisati daljše ukaze, ampak programirajo z nizanjem blokov. Ta način programiranja je zelo enostaven in primeren za učenje mlajših učencev. Programski jezik vsebuje obsežno zbirko sličic in zvokov ter je idealen za ustvarjanje iger in animacij (Vorderman, 2017). Skozi ustvarjanje projektov se v Scratchu naučijo osnovnih računalniških konceptov: zaporedje ukazov, dogodki, sočasnot, zanke, pogojni stavki, operaterji in podatki (Koron, 2021). Z učenjem programiranja učenci razumejo, kako podajati ukaze, da se figure premikajo, kako uskladiti več ukazov skupaj (sinhronizacija) ter znajo strateško in smiselno zapisati zaporedje gibanja.

Čeprav se računalniško mišljenje kot kognitivna spretnost tesno povezuje z računalniškim programiranjem, to še zdaleč ni edino področje oz. dejavnost, ki zahteva uporabo procesov računalniškega mišljenja. Prepoznavamo ga tudi na področju glasbenega komponiranja.

2.2 POVEZAVA MED RAČUNALNIŠKIM MIŠLJENJEM IN GLASBENO KOMPOZICIJO

Povezovanje glasbenega ustvarjanja in računalniškega mišljenja daje učencem nov pogled na obe temi. Učenci spoznajo, da je glasbeno komponiranje zelo podobno računalniškemu programiranju. V obeh primerih gre za ustvarjanje točno določenih korakov, ki si morajo med seboj slediti v točno določenem vrstnem redu, da glasba zveni točno tako, kot želimo. Naučijo se, da glasba temelji na natančno določenih pravilih (lestvice, ritem, akordi, glasene oblike) (Liu in Ting, 2017). Glasbo in programiranje povezuje tudi to, da morata biti oba izvedena, da imata želeni učinek. Če inštrument samo postavimo v prostor tako kot sliko, še ne bomo slišali glasbe. Potrebujemo nekoga, ki ustvarjeno kompozicijo z igranjem prenese na inštrument. Tudi računalnik se ne odziva sam, ampak potrebuje napisan program, ki bo pričel delovati, ko mu bomo dali ukaz. Predvsem pa sta danes tako glasba kot programiranje neločljivo povezana s tehnologijo. Adrian Jewell (2019) pravi, da je vadenje klavirja enako kot reševanje algoritma.

3 PRIMER PROGRAMIRANJE PRI POUKU GLASBENE UMETNOSTI

3.1 NAČRTOVANJE UČNE URE

Primer učne ure s programiranjem smo izvedli s šestošolci, saj so ti v šolskem letu 2024/2025 izbrani kot inovativni oddelek, s katerim v okviru projekta Inovativna pedagogika 5.0 intenzivneje razvijamo digitalne kompetence in temeljne vsebine RIN. V mesecu decembru smo izvedli tehniški dan, ob katerem so bolje spoznali program Scratch ter ustvarili svoj prvi izdelek. Program Scratch pogosto uporabljajo tudi pri uri likovne umetnosti, tako so bili učenci že precej veščji dela s tem programom. Zato smo se odločili, da ta način dela vključimo še pri pouku glasbene umetnosti, predvsem z namenom, da bi motivirali učence za ustvarjanje in poustvarjanje z notnim zapisom. Z izvedbo učne ure smo želeli preveriti slednje:

1. Kako vpliva programiranje na motivacijo učencev za učenje in ustvarjanje?
2. Ali je programiranje učinkovita metoda za utrjevanje ritma in tonske abecede?
3. V kolikšni meri imajo učenci že razvito računalniško mišljenje?

Učne aktivnosti smo oblikovali na podlagi opredeljenih učnih ciljev. Učenci:

- utrdijo in ponovijo ritmične vrednosti ter jih samostojno uporabljajo
- ponovijo in utrdijo tonsko abecedo
- ustvarijo lastno melodijo
- za svojo melodijo ustvarijo notni zapis
- za melodijo ustvarijo programsko kodo v programu Scratch
- za notni zapis lastne melodije ustvarijo animacijo v programu Scratch

Pri izvedbi ure z uporabo IKT je pomembno, da zagotovimo zadostno število računalnikov ali tabličnih računalnikov. Prav tako je pomembno, da učitelj dobro pozna program, v katerem bodo učenci ustvarjali, da lahko učencem nudi primerno vsebinsko in tehnično pomoč.

3.2 POTEK UČNE URE

Učenci so ustvarjali v manjših skupinah treh učencev. Vsaka skupina je prejela list z notnim črtovjem, melodični inštrument ter tablico. Učenci so se uvodoma seznanili s cilji učne ure in z navodili za delo. Za izvedbo so imeli na voljo tri šolske ure.

Nato smo skupaj določili kriterije uspešnosti, ki so omogočali učencem sprotno vrednotenje njihovega dela. Kriteriji so bili:

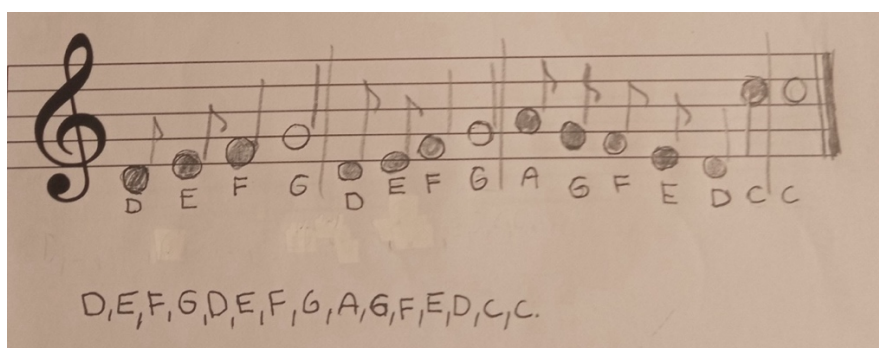
1. Ustvarili smo zanimivo kratko melodijo.
2. V programu Scratch smo ustvarili kodo za svojo melodijo.
3. Ustvarili smo animacijo notnega zapisa, ki je sinhronizirana s kodo zvočnega zapisa.

Učenci so znotraj skupine najprej planirali posamezne korake in si porazdelili delo.

Ustvarjanje melodije in zapis algoritma na skicirni list

Pri ustvarjanju melodije so si pomagali s ksilofonom, zvončki in klavirjem. Ustvarjanje po posluhu ni povzročalo težav. Imeli so veliko idej in kmalu so oblikovali melodijo, ki je bila všeč vsem članom skupine. Pri tej nalogi jih je bilo treba opomniti, da so časovno omejeni in da je smiselno, da ne pretiravajo z dolžino melodije, saj to pomeni, da bo programiranje potekalo toliko dlje.

Vsak ton melodije so učenci zapisali s tonsko abecedo in s tem postavili osnovo za programsko kodo. Melodijo so nato zapisali še z notami.



Slika 1: Primer zapis melodije z notami in tonsko abecedo

Programiranje v programu Scratch

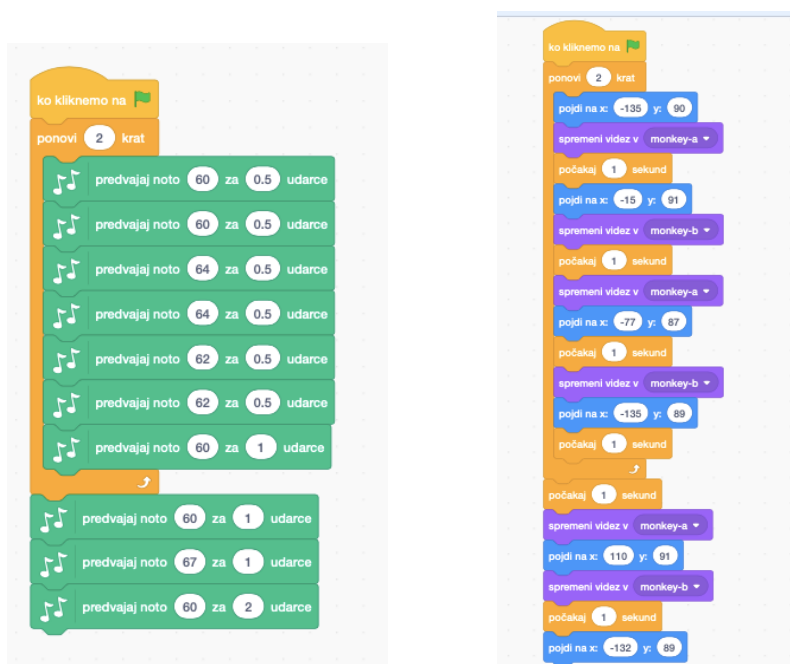
Zaporedje tonov ustvarjene melodije so učenci prenesli v program Scratch. Pri programiranju melodije jim je največ težav povzročalo določanje ritma oz. pravičen zapis trajanja posameznega tona. Pri tem so si večinoma znali pomagati z logičnim matematičnim računanjem, pri čemer ima vsaka nota jasno določeno vrednost (tabela 1). Večje kot je število, daljša je nota.

Tabela 1: Primer določanja ritma s števili v Scratchu

osminka	0,25
četrtnika	0,5
polovinka	1
celinka	2

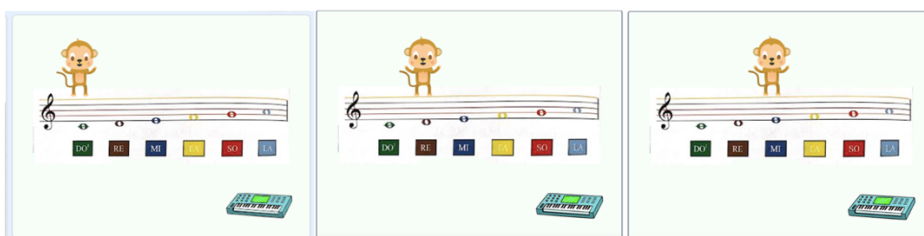
Tisti, ki niso obvladali razmerij med notnimi vrednostmi, pa so ritem določali z večkratnim poslušanjem in popravljanjem trajanja tona po posluhu.

Ko so kodo melodije uspešno zapisali, so se lotili ustvarjanja animacije notnega zapisa. Ustvariti so morali programsko kodo za animacijo figure, zanjo ustvariti več različnih videzov in uskladiti gibanje z glasbo. Spodnja slika prikazuje primer sinhronizacije kodnega zapisa zvoka in premikanja figure.



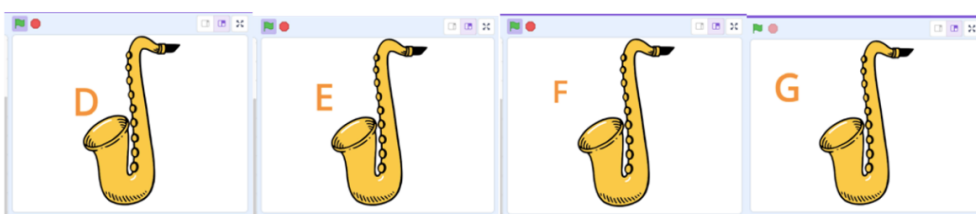
Slika 2: Programska koda za melodijo (levo), programska koda za premikanje figure (desno)

Največ časa so učenci porabili za iskanje rešitev, kako v animaciji prikazati notni zapis. S svojo iznajdljivostjo in vztrajnostjo so našli izvirne rešitve, kako čim bolj preprosto in učinkovito ustvariti animacijo. Nekateri so za osnovno prikaz izbrali lestvico in s premikanjem figure označili zaporedje tonov (slika 3).



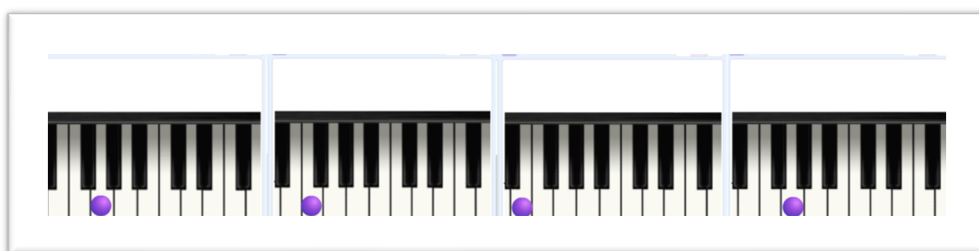
Slika 3: Prikaz različnih postavitev figure (opica)

Drugi so melodijo zapisali z imeni tonov kot prikazuje slika 4.



Slika 4: Prikaz različnih videzov za video animacijo

Ena skupina pa je melodijo predstavila s pomočjo klaviature (slika 5).



Slika 5: Prikaz različnih postavitev figure (žogica)

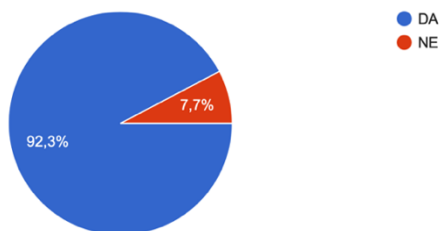
4 REZULTATI

Rezultate primera smo pridobili z metodo študije primera z opazovanjem, analizo izdelka ter anketo. Med izvajanjem aktivnosti smo analitično spremljali delovni proces, preverjali, kako učenci med seboj sodelujejo in kako uspešni so pri ustvarjanju. Vsa opažanja smo sproti zapisali in jih povzeli v spodnjih odstavkih skupaj z rezultati ankete, ki so jo učenci rešili po opravljeni nalogi.

1. Kako vpliva programiranje na motivacijo učencev za učenje in ustvarjanje?

Učenci so bili skozi učni proces ves čas aktivni. Le občasno je kakšen posameznik potreboval dodatno spodbudo za sodelovanje. Takrat smo sošolce prosili, da so učenca aktivneje vključili v proces dela. Ugotovili smo, da v veliki večini učenci zelo radi programirajo in pri tem ne potrebujejo veliko spodbude za sodelovanje. Svoje navdušenje nad programiranjem so izrazili tudi ob koncu šolske ure, ko smo izvedli kratko refleksijo, v anketi pa so pričakovano izrazili navdušenje nad nalogo in svojim izdelkom.

A) Ali si ponosen/ponosna na svoj končni izdelek?



Slika 6: Prikaz odgovorov učencev

Vsi učenci, razen ene izjeme, so bili na svoj izdelek ponosni.

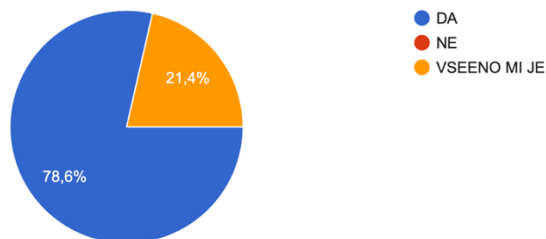
B) Pojasni, zakaj si ponosen/ponosna oz. zakaj nisi?

Tabela 2: Odgovor učencev na kaj so oz. niso ponosni

Ker smo sodelovali skupaj in ga uspeli narediti zelo dobro.	Ponosna sem zato ker so to naredile same brez kakšnem napake ali nekega velikega problema.
Ker zgleda lepo.	Zato ker, imamo zelo dobro melodijo
Ker sem prvo mislila da bo zelo težka naloga in ko sem jo preizkusila in končala je bilo odlično.	Ker smo se zelo potrudili
Ker nam je dobro uspelo.	Ker smo se veliko trudili.
ker je izpadlo lepo.	Zato ker smo naredili skladbo ki gre v uho in zanimivo animacijo.
Ker je lep in zelo mi je všeč meodija.	Ker smo se z sošolci veliko motili.
Ker smo ustvarile lepo melodijo.	

Učenci so najbolj ponosni na dobro sodelovanje v skupini in na svoj trud, ki so ga vložili v izdelek. Nekateri so bili ponosni tudi na melodijo, ki so jo ustvarili s pomočjo glasbila.

- C) Ali si želiš še več tovrstnih ustvarjalnih nalog s pomočjo tabličnega računalnika pri glasbeni umetnosti?



Slika 7: Interes učencev za pouk s tabličnim računalnikom

Učenci so v veliki večini izkazali navdušenje nad vključevanjem tabličnega računalnika pri glasbeni umetnosti.

2. Ali je programirnje učinkovita metoda za utrjevanje ritma in tonske abecede?

Pri uri glasbe smo pri enaki temi pri učenju brez računalnika zelo težko motivirali učence za poustvarjanje in ustvarjanje z notnim zapisom. Tokrat pa se učenci niso pritoževali nad zahtevnostjo učne snovi. Tako zelo so bili veseli ustvarjanja v Scratchu, da jih izbrana učna snov sploh ni omejevala pri aktivnem sodelovanju. Pri sestavljanju programske kode so ves čas aktivno ustvarjali z ritmom in s tonskimi višinami na njim zabaven način in tako učinkovito razvijali svoje glasbeno znanje in posluh. Ugotovili smo, da učenci izjemno radi programirajo ne glede na dano temo. Po uri so učenci v anketi morali zapisati imena tonov C-dur lestvice, določiti število tonov v lestvici ter poimenovati notne vrednosti. Vsi učenci so podali pravilne odgovore in s tem potrdili, da so ta učni cilj v celoti dosegli ter potrdili, da je programiranje lahko zelo učinkovita metoda za utrjevanje glasbenih vsebin.

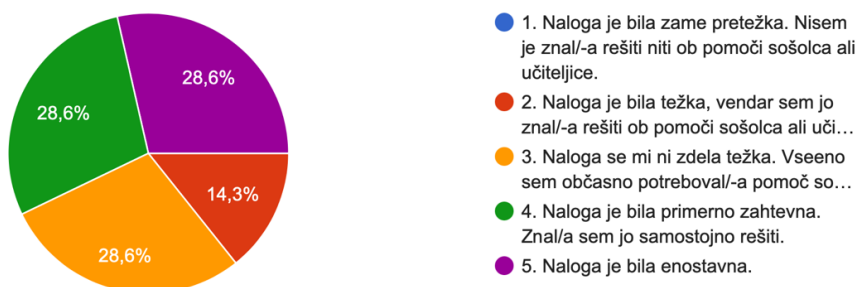
3. V kolikšni meri imajo učenci že razvito računalniško mišljenje? Ali bodo znali samostojno ustvariti pravilno programsko kodo za glasbo ter jo sinhronizirati z animacijo? Kako ustvarjalni bodo?

Učenci, ki imajo razvito računalniško mišljenje, znajo dani problem razdeliti na manjše korake in poiskati rešitev v obliki konkretnih navodil, zapisanih v programskem jeziku (Vorderman, 2017). V tej skupini so učenci, ki so učno uspešnejši, hitreje prišli do konkretnih rešitev in nalogo končali uspešno. Vendar iz oblikovnega, vizualnega vidika ta ni odstopala od animacij tistih, ki so za delo potrebovali več časa. Najbolj so presenetili tisti učenci, ki so učno šibkejši, a imajo razvito glasbeno ali likovno izraznost. Ti so do rešitev prihajali počasneje, a so bili končni izdelki vizualno-grafično bolj izvirni. Tako smo

prišli do ugotovitve, da imajo učenci z boljšimi intelektualnimi sposobnostmi bolj razvito računalniško mišljenje. Hitreje in bolj samostojno znajo reševati logične probleme in zastavljene izzive. A vseeno potrebujejo vsi učenci, tudi najbolj uspešni, še veliko vaje in učnih izkušenj, da bodo razvili računalniško mišljenje v največji možni meri.

Spodaj so predstavljeni vtisi učencev o zastavljenih nalogah.

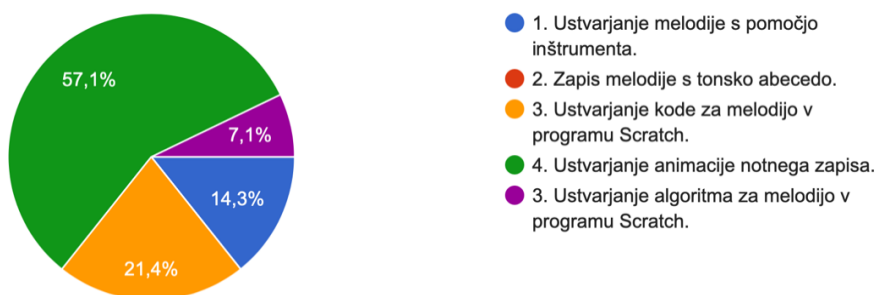
A) Kako bi ocenil/a težavnost naloge ustvarjanja v Schrathu od 1 do 5?



Slika 8: Prikaz odgovorov učencev o težavnosti naloge

Rezultati kažejo, da je slaba tretina učencev (28,6%) nalogo ocenila kot enostavno, enak odstotek učencev pa kot primerno težko. Oboji so nalogo znali opraviti povsem samostojno. Enak delež učencev (28,6%) je nalogo prav tako ocenil kot primerno težko, a so potrebovali pomoč sošolca ali učitelja. Le 14,3 odstotka učencev je nalogo ocenilo kot težko in so pri svojem delu ves čas potrebovali pomoč učitelja ali sošolca.

B) Kateri del naloge se ti je zdel najtežji?



Slika 9: Prikaz odgovorov učencev, kateri del naloge jim je predstavljal največ težav

Največ učencev je kot najtežji del naloge izpostavili ustvarjanje animacije notnega zapisa. Drugi najpogostejši odgovor v precej manjšem deležu pa je bil ustvarjanje kode za

melodijo v programu Scratchu. Vseeno pa so posamezniki izpostavili kot najtežji del naloge ustvarjanje melodije s pomočjo instrumenta. Svoje utemeljitve so opredelili v naslednji tabeli (tabela 3).

C) Pojasni, zakaj!

Tabela 3: Težave s katerimi so se učenci soočali pri ustvarjanju

Zato, ker tablica občasno ni delovala.	Ker se nam je 1 zbrisalo brisalo
Ker ni veliko elementov za vstaviti	Ker nisem znal narediti takšne animacije
Zato ker nisem vedela kako začeti.	Ker smo se velikokrat zmotili.
Ker je bilo s tem največ dela.	Zato ker moraš poiskati kje je kaj
Zato, ker smo rabili narediti da se note pojavijo in to nam je bilo težko.	ker moraš vsako odmevanje določiti kako dolgo bo trajalo
Ker je težko narediti da so note enako velike.	Ker nismo znali vstavi slike iz interneta.
Ker je bilo težko izmeriti kdaj pride katera nota in potem to ustvariti.	Zato, ker se nam je velikokrat kaj zbrisalo ali pa premaknilo ampak vseeno ni bilo tako težko

Učenci so največ težav imeli pri programiranju notenga zapisa v programu Scratch. Kot najpogostejši razlog so navedli pomanjkanje znanja pri oblikovanju različnih vizualnih podob figur, vstavljanju slik v program Sctrach in z usklanjevanje animacij s tempom melodije.

5 ZAKLJUČEK

V naši raziskavi smo pri predmetu glasbena umetnost preverili, kako programiranje vpliva na motivacijo učencev za učenje in ustvarjanje ter kako učinkovita je ta metoda dela za učinkovito utrjevanje tonske abecede in osnovnih ritmičnih vrednosti v šestem razredu. Ugotovili smo, da učenci zelo radi programirajo ne glede na dano učno temo, zato so veliko bolj odprti za usvajanje novih znanj. Prav tako smo preverili v kolikšni meri že imajo učenci razvito računalniško mišljenje in ali bodo znali poiskati rešitve za dane probleme. Nalogo je samostojno uspešno rešilo več kot polovica učencev, ostali so pri svojem delu potrebovali pomoč sošolcev ali učitelja. Rezultati te študije primera nam potrjujejo, da lahko učitelji s pomočjo programiranja v enostavnih programskih jezikih, učencem omogočijo učinkovito pridobivanje novega znanja in razvijanje glasbene ustvarjalnosti in radovednosti pri učencih. Računalniško mišljenje se ne razvije z enkratno aktivnostjo programiranja v šoli, ampak je potrebno vsebine RIN sistematsko vključevati v pouk

tekem celega šolskega leta. Res je, da imamo osnovnošolski učitelji na tem področju malo znanja in izkušenj, a pomembno je, da se odločimo in pričnemo to področje raziskovati ter ga postopoma uvajamo v pouk. Program Scratch je primeren za začetno učenje programiranja in omogoča ustvarjanje raznolikih interaktivnih animacij in iger. Pomembno je samo, da so naloge vsebinsko enostavne, saj so učenci časovno omejeni. Za programiranje potrebujejo dovolj časa, dajo uspešno končajo, kar je ključno za uspešno napredovanje in razvijanje pozitivnega odnosa do učenja. V prispevku smo potrdili, da se da glasbo zelo smiselno in učinkovito povezovati s področjem računalništva. Zato si želimo, da bi primer dobre prakse spodbudil druge učitelje glasbe, da povežejo glasbene aktivnosti s programiranjem.

LITERATURA

- Holcar, A., Borota, B., Breznik, I., Jošt, J., Kerin, M., Kovačič, A., Pesek, A. (2011). Učni načrt: Program osnovna šola. Glasbena vzgoja. Ljubljana: Ministrstvo za šolstvo in šport: Zavod RS za šolstvo. https://www.gov.si/assets/ministrstva/MVI/Dokumenti/Osnovna-sola/Ucni-nacrti/obvezni/UN_glasbena_vzgoja.pdf
- Jewell, A. (26. 3. 2019). *Music and codin: All the ways are similar*. App Academy. <https://www.appacademy.io/blog/state-management-observations-on-how-music-and-coding-are-so-similar>
- Koron, A. (2021). *Razvoj računalniškega mišljenja skozi Scratch projekte: magistrsko delo*. <http://pefprints.pef.uni-lj.si/id/eprint/6710>
- Vorderman, C. (2017). *Računalniško programiranje za otroke*. Mladinska knjiga.
- Liu, C.-H., in Ting, C.-K. (2017). Computational Intelligence in Music Composition: A survey. IEEE Transactions on Emerging Topics in Computational Intelligence, 1(1), 2-15. <https://doi.org/10.1109/tetci.2016.2642200>
- Stefanija, L., Bonin, M., Beuermann, D., in Holcar, A. (2016). O koristi in škodi teorije glasbe v glasbenem izobraževanju na osnovnih in glasbenih šolah. *Javno glasbeno šolstvo na Slovenskem: pogledi ob 200-letnici*, 25, 203–216.
- Krajnc, R., Košir, K., in Čotar Konrad, S. (2017). Računalniško mišljenje - kaj je to in zakaj bi ga sploh potrebovali? *Vzgoja in izobraževanje*, 48(4), 9–19. https://www.zrss.si/wp-content/uploads/2023/06/02_RadovanKrajnc-idr-1.pdf
- Skupina RINOS. (b. d.). *Brez nepotrebnega sedenja pred zasloni. Računalništvo in informatika za vse*. <https://www.racunalnistvo-in-informatika-za-vse.si/manjcasapredzaslonom.html>