

Evaluacija WEB aplikacije za učenje osnove računalstva

Poropat, Melita

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:616793>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-02**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli

Fakultet informatike

Melita Poropat

EVALUACIJA WEB APLIKACIJE ZA UČENJE

OSNOVA RAČUNALSTVA

Diplomski rad

Pula, 2018.

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

Melita Poropat

EVALUACIJA WEB APLIKACIJE ZA UČENJE

OSNOVA RAČUNALSTVA

Diplomski rad

JMBAG: 2424013378, izvanredan student

Studijski smjer: Informatika, nastavni smjer

Predmet: Mobilne aplikacije

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacije znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentori: doc.dr.sc. Sovilj Siniša

doc.dr.sc. Etinger Darko

Pula, svibanj 2018.



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana **Melita Poropat**, kandidat za magistra edukacije informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, _____, _____ godine



IZJAVA
o korištenju autorskog djela

Ja, **Melita Poropat** dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom **Evaluacija web aplikacije za učenje osnova računalstva** koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

SADRŽAJ

UVOD	1
WEB APLIKACIJA CODE.org®	2
Primjena Code.org tečajeva računalnih znanosti u školama u svijetu	3
Opis web aplikacije Code.org	3
Definiranje sposobnosti kodiranja.....	6
Matrica koncepata programiranja i težine tečajeva	6
Određivanje stručnosti učenika: Koncepti + Težina	11
PROJEKTNA NASTAVA	13
Povijesni pregled projektne nastave	14
Primjeri provedenih aktivnosti sa učenicima 1. razreda	15
Aktivnost 1 – Moja jutarnja rutina – usvajanje pojma algoritam.....	15
Aktivnost 2 – Kako prijeći cestu kao robot?	15
Aktivnost 3 – Igra Školice na ploči	16
Aktivnost 4 – Povuci – spusti tehnika slaganja blokova	16
Aktivnost 5 – Ljuta ptica u programiranju	17
Aktivnost 6 - Predstavljam svoj rad	18
Aktivnost 7 – Ljuta ptica u programiranju	19
Aktivnost 8 – Predstavljam svoj rad	20
Aktivnost 9 – Programeri kao umjetnici	20
Aktivnost 10 – „Ljudski robot“ crta kvadrat	21
Aktivnost 11 – Izgradi piramidu.....	22
Aktivnost 12 – Crtamo likove	22
Aktivnost 13 – Ja kao umjetnik.....	23
Aktivnost 14 - Pas i mačka razgovaraju o Pokemonima	23
Primjeri provedenih aktivnosti sa učenicima 3. razreda	24
Aktivnost 1 – Algoritmi	24
Aktivnost 2 – Programiramo ili se igramo?	26
Aktivnost 3 – Zapetljancije	27
Aktivnost 4 – Petlje.....	27
Aktivnost 5 – Uvjeti i odluke u programiranju	28
Aktivnost 6 – Veliki problem > mali problem > nema problema!	29
Aktivnost 7 – Zombi i suncokret u labirintu	30
Aktivnost 8 – Pčela i funkcije	32

Aktivnost 9 – Kako otkriti skriveno?	34
Aktivnost 10 – Ugniježdene petlje.....	36
Aktivnost 11 – Ispravimo pogreške	37
Aktivnost 6 – Primijenimo naučeno u programiranju	38
Aktivnost 12 – Roboti u 3. razredu	40
Aktivnost 13 – Funkcijski blok	42
Aktivnost 14 – Zupčanici u 3. razredu	43
Aktivnost 15 – Ispitajmo uvjete	45
ISTRAŽIVANJE	47
TAM model kao okvir istraživanja	47
Nadogradnja TAM modela	49
Zadovoljstvo	49
Varijable modela	50
Metodologija istraživanja	50
Hipoteze istraživanja	51
Sudionici istraživanja	51
Istraživački instrumenti i prikupljanje podataka	51
Analiza podataka	52
Deskriptivna statistika	52
Model procjene uspješnosti	53
Rezultati istraživanja	53
Konstrukcija i validacija instrumenata istraživanja	53
ZAKLJUČAK.....	56
LITERATURA.....	58
POPIS SLIKA	60
POPIS TABLICA.....	61
PRILOG.....	62
Evaluacijski listić	62
SAŽETAK.....	64
ABSTRACT	65

UVOD

Danas su djeca okružena digitalnim tehnologijama i obrazovnim web aplikacijama koje nisu bile dostupne prije nekoliko godina. Međutim, nema dovoljno vremena ili resursa za procjenu svake aplikacije koja dođe na tržište. Ovaj rad nudi jedan od mogućih načina definiranja potencijala obrazovnog učinka web aplikacije za učenje osnova računalstva, tzv. kodiranja. U radu se opisuju rezultati istraživanja kojeg sam provela tijekom tekućeg obrazovnog programa školske godine 2017./2018. Algebra Digitalne akademije u Hrvatskoj. Algebra ustanova za obrazovanje u ovom programu provodi suvremene prakse u vezi s digitalnim materijalima koja obuhvaćaju i STEM znanstveno, tehnička, inženjerska i matematička područja (engl. *Science, Technology, Engineering and Mathematics*) učenja za učenike osnovnih škola.

Cilj ovog rada bio je prilagodba istraživačkog modela koji opisuje stavove ka budućoj upotrebi i zadovoljstvu korištenjem Code.org web aplikacije za učenike osnovnih škola u Hrvatskoj, temeljenog na modelu prihvaćanja tehnologije TAM (engl. *Technology acceptance model*) kojeg je razvio Fred Davis. Upitnik za evaluaciju web aplikacije Code.org kreiran je kako bi ispitali stavove prihvaćanja učenika prema korištenju web aplikacije i njihovo zadovoljstvo korištenjem web aplikacije. Podaci 114 učenika osnovnih škola (8-14 godina) prikupljeni su u privatnoj školi Algebra Digitalna akademija u nekoliko gradova Republike Hrvatske. Model se sastoji od četiri konstrukta: stavovi prema upotrebi, zadovoljstvo, percipirana korisnost i percipirana jednostavnost korištenja. Model je procijenjen korištenjem modela strukturne jednadžbe SEM (engl. *Structural equation modeling*) za koji je korišten softver SmartPLS3. Zadovoljstvo i stav prema upotrebi pokazali su se odlučujući čimbenici u predviđanju daljnje upotrebe tehnologije, a percipirana korisnost bila je značajna za pokazivanje pozitivnih stavova ka korištenju web aplikacije prilikom učenja kodiranja. Izlazni rezultati analize pokazuju da model u cjelini dobro odgovara podacima i ukazuje na značajne odnose između varijabli u modelu. Rezultati potvrđuju da je TAM vrijedan alat za predviđanje stavova i zadovoljstva uporabom. Također, pokazuju da je učenikova percipirana korisnost najviše pridonijela pozitivnim stavovima i općenitom zadovoljstvu korištenja web aplikacije Code.org za učenje osnova računalstva.

WEB APLIKACIJA CODE.org®

Code.org® je neprofitna organizacija koja je posvećena širenju informatičke znanosti u školama i povećanju sudjelovanja žena i nedovoljno zastupljenih manjina. Njihova vizija je da svaki učenik u svakoj školi treba imati priliku učiti računalne znanosti, baš kao i biologiju, kemiju ili matematiku. Code.org organizira godišnju kampanju sata programiranja „Hour of Code“ koji je dosad uključio 10% svih učenika u svijetu i osigurava vodeći kurikulum za računalnu znanost K-12 u najvećim školama u Sjedinjenim Državama. Code.org su pomogli kreirati donatori, kao što su to Amazon, Facebook, Google, Infosys Foundation i Microsoft.

Tablica 1. Code.org ciljevi i postignuća

Code.org Ciljevi	Postignuća
Poboljšati raznolikost u tečajevima osnova računalnih znanosti CS (engl. <i>Computer Science</i>)	Na tečajevima, 45% studenata su djevojčice, a 48% su manje zastupljene manjine. U srednjoškolskim online učionicama 37% je djevojaka, 56% afroameričkih ili latinoameričkih učenika.
Inspirirati studente	Deseci milijuna učenika diljem svijeta okušali su se na Satu kodiranja. (561.338.257 prijavljenih, 49% žena)
Kreirati korisne tečajeve	99% anketiranih nastavnika preporučuje Code.org
Ući u učionice diljem svijeta	813.197 učitelja su se prijavili za podučavanje naših uvodnih tečajeva na Code Studio i upisano je 27.145.973 studenata.
Pripremiti nove učitelje za računalstvo	Obučili su preko 72.000 novih nastavnika za podučavanje CS-a po razredima K-12 (osnovno-školsko obrazovanje do 12. razreda u nekim zemljama svijeta).
Promijeniti nastavni plan i program u kurikulumu	Partneri su oko 180 od najvećih školska kako bi dodali CS u nastavni plan i program. Poučavaju gotovo 10% svih učenika SAD-a i 15% latino i afroameričkih učenika.
Postaviti pravila za podršku računalstvu u školama	Politike su se promijenile u više od 40 država SAD-a kako bi se uspostavili standardi obrazovanja za računalstvo, te da se tečajevi CS-a okreću prema srednjoj školi
Globalizacija	Tečajevi su dostupni na više od 50 jezika, koji se koriste u 180 zemalja.

Primjena Code.org tečajeva računalnih znanosti u školama u svijetu

U samo 4 godine, 25 zemalja, 40 američkih država i gotovo 200 američkih gradova i škola najavili su planove za proširenje i raznolikosti računalnih znanosti kao dio osnovnog i srednje školskog obrazovanja; više od 72 000 učitelja SAD-a prisustvovalo je Code.org radionicama za podučavanje računalne znanosti; 750.000 učitelja počelo je koristiti Code.org kako bi podučavalo nastavu iz računalnih znanosti za preko 25 milijuna studenata širom svijeta; raznolikost u učionicama računalnih znanosti poboljšana je četiri godine zaredom; a sat kodiranja prešao je 500 milijuna korisnika – što bi se moglo rastumačiti da je za svakih 10 učenika diljem svijeta po jedan učenik sudjelovao na satu programiranja.

Opis web aplikacije Code.org

Program Code.org računalnih znanosti sastoji se od pet tečajeva usmjerenih na učenike osnovnih škola. Tečajevi 1-4 su dizajnirani tako da imaju oko ukupno 20 školskih sati po razredu, uključujući i aktivnosti koje su video lekcije (engl. *unplugged*). Mnogo materijala osmišljeno je za učitelje u učionici. Obično se uči jednom ili dva puta tjedno tijekom semestra.

Primarna značajka tih tečajeva su slaganje zagonetke uporabom kodnih blokova koje vode učenike kroz korištenje određenog programskog koncepta (poput petlje) za rješavanje malog problema. Zagonetke su slične onima poznatim pod nazivom "*Parson's Puzzles*" u kojima se postavlja mali izazov ili zagonetka, a rješenje se određuje slaganjem skupova blokova koda kao alatom za rješavanje tog problema. Taj pristup programiranju učenja onemogućuje određene vrste tipičnih "početničkih grešaka" povezanih sa sintaksom i, u teoriji, dopušta kognitivnom opterećenju učenika da se više usredotoči na upotrebu koda kao medija za rješavanje problema (Parsons i Haden, 2006.).

Tečajevi temeljeni na računalnoj znanosti razvrstavaju se u stupnjeve tj. svaka faza je jedna lekcija ili blok školskih sati, a svaka faza sastoji se od niza "razina" - obično 10-15 po stupnju. Većina razina su razine kodiranja (koje nazivamo zagonetke za ovaj rad). Postoji i niz lekcija posvećenih i ostalim aktivnostima, a među njima su i kviz pitanja i odgovora poput kviza sa mogućnošću slaganja podudaranja ili višestrukog izbora. Ispod je navedena tablica koja sažima tečajeve, pokazujući koliko stupnjeva i

razina svaka ima zajedno s programskim konceptima koje učenici susreću tijekom tečaja preuzeta sa godišnjeg izvještaja Code.org za 2017. godinu.

Tablica 2. Prikaz CS osnovnih tečajeva i njihova namjena i koncepti

CS osnovni tečajevi	Kome je namijenjen?	Koncepti
Tečaj 1 18 nivoa 119/146 razina kodiranja	Ne-čitači (K-1) Napomena: ovaj tečaj nije korišten pri mjerenju učeničke sposobnosti uporabe	Upute o algoritmima, petlje
Tečaj 2 19 nivoa 128/159 razina kodiranja	Učenici osnovne škole Preporučeno je početi od 2 do 5 razreda za one učenike koji se nisu susreli sa računalnim znanostima prije	Algoritmi, petlje (manja količina uvjeta i događaja)
Tečaj 3 21 nivoa 148/177 razina kodiranja	Učenici osnovne škole Osmišljeno za učenike od 3-5 razreda koji su završili Tečaj 2	Algoritmi, petlje, uvjeti, događaji, funkcije
Tečaj 4 22 nivoa 161/167 razina kodiranja	Učenici osnovne škole Osmišljeno za učenike od 5-6 razreda koji su završili Tečaj 3	Algoritmi, veći broj petlji, uvjeti, događaji, funkcije sa parametrima, varijable
Ubrzani tečaj 20 nivoa 109/109	Učenici srednje škole Namijenjeno učenicima srednjih škola bez prethodnog predznanja računalnih	Algoritmi, petlje, uvjeti, događaji, funkcije

razina kodiranja	znanosti. Tečaj pokriva većinu materijala iz 2 i 3 tečajeva	
------------------	---	--

Izvor: Code.org 2017 godišnji izvještaj, <https://code.org/about/2017>

Većina učenika koji koriste ove tečajeve su u osnovnoj školi i kodiraju po prvi put. Dakle, dok tečajevi upućuju učenike da riješe zagonetke pomoću vrlo osnovnih koncepata kodiranja, uz anegdote, učitelji i stručni edukatori uče učenike osnovama računalstva navodeći ih da slijede upute, koriste petlje, uvjete, događaje, funkcije i varijable.

Tečajevi Osnova računalstva osmišljeni su kako bi pomogli učenicima naučiti koristiti osnovne koncepte kodiranja kao što su slijed, iteracija i uvjetovana logika, ali i procjene.

Svaki put kad učenik riješi zagonetku Code Studio bilježi informacije: vremenske oznake za pokretanje, pokušaj i završetak slagalice; učenikov kod iz svakog pokušaja; koji upućuju na pregled, i tako dalje. Iako se ne može sa točnošću utvrditi je li učenik zaista nešto naučio ili o tome što je uzrokovalo naučeno na ovaj ili onaj način ipak možemo pratiti napredak učenika. Postoji puno varijabli koje mogu utjecati na to - učenici koji rade zajedno, interakcije sa učiteljem, dodatne domaće zadaća, pomoć roditelja, izravna nastava, pravi razred, slobodno vrijeme igranja kod kuće - stoga ne postoji točan uvid u cjelokupnu situaciju i nikada neće ni moći biti pod potpunom kontrolom. No ako gledamo na zagonetke kao da su individualni test ili kviz pitanja možemo vidjeti ako ih je učenik dobro riješio.

Definiranje sposobnosti kodiranja

Što znači sposobnost "kodiranja"? Ukratko to znači da učenik može primijeniti kodne koncepte (algoritmi, iteracije, funkcije, itd.) kako bi riješio niz postavljenih zagonetki koristeći kodne blokove na Code.org platformi.

Matrica konceptata programiranja i težine tečajeva

Code.org nudi suvremeni pristup pokušaja mjerenja učeničkog "znanja" u nastavnom planu i programu Osnove računalstva gdje su identificirali šest široko postavljenih konceptata programiranja koji se obrađuju kroz: algoritmi, petlje, uvjeti, događaji, funkcije i varijable. Također, razbili su neke pojmove u nekoliko opisnih dijelova. Na primjer, "petlje" (stenografija za: iteraciju) predstavlja prilično veliki koncept koji se u Code.org okruženju prikazuje na nekoliko različitih načina: ponavljanje petlje, ponoviti sve do x, for petlje i tako dalje. Slično tome, razbili su funkcije za razlikovanje pomoću funkcija s parametrima. Skup tih pojmova prikazan je u slijedećoj tablici.

Tablica 3. Prikaz CS konceptata iz programiranja u Code.org

Sekvence (Algoritmi)	Uvjeti	Varijable	Petlje	Funkcije	Događaji
			<ul style="list-style-type: none">• Ponovi• Ponovi "until" /Ponavljanje "while"• For Petlje	<ul style="list-style-type: none">• Definicija i Pozivanje• Definiranje i Pozivanje ili/uvođenje Parametara	

Izvor: Code.org 2017. godišnji izvještaj, <https://code.org/about/2017>

Tečajevi također predstavljaju pojmove kao što su ispravljanje pogrešaka, suradnja, sigurnost na internetu i još mnogo toga, no ti su koncepti i lekcije često "isključeni" ili se događaju u učionici izvan okruženja Code Studio web platforme. Mjerni podaci prikazani ovdje samo mjere izvedbu učenika na zagonetkama u Code Studio-u koji se odnose na računalno programiranje.

Ocjenjivanje rješenja zagonetki po "težini" vrši se po ljestvici od 1 do 5 točaka, od kojih je 1 "lagano" a 5 "teško". Općenito govoreći, ocjena 1 značila je slagalicu koja zahtijeva praktično nikakvo rješavanje problema - jednostavno bi moglo zatražiti od učenika da

klikne gumb i promatra rezultate ili pogleda video lekciju. Ocjena 5 značila je da je slagalica bila vrlo izazovna, ili je rješenje zahtijevalo sofisticiranu primjenu koncepata kako bi se postiglo pravilno ili proizvelo optimalno rješenje.

Slijedi tablica pojma iz računalstva i težinska procjena računalnog koncepta. Teoretski, svaka je ćelija sadržavala popis stavki, tipičnih za Code.org zagonetke, za koje se misli da se podudaraju s obzirom na koncept i težinu. Slijedi tablični isječak koji prikazuje razliku između poteškoća 1 i 5 za ponavljanje petlje, tzv. “repeat” petlje.

Tablica 4. Prikaz težine i opis koncepta na pripadajućoj težinskoj razini

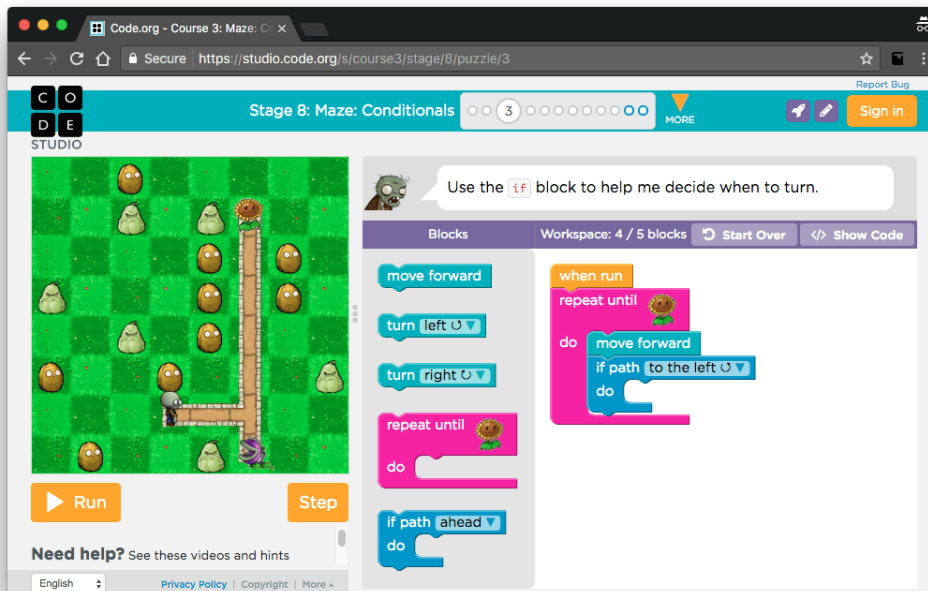
Težina	Koncept: Repeat Petlje
1	a) Premjestite postojeći kôd u petlju s skočnom porukom b) Izradite jednostavnu (1-instrukcijsku) petlju s porukom c) Utvrdite područja programa koji se ponavljaju točno d) Izmijenite ili izbrišite instrukciju s jednom petljom sa skočnom porukom e) Promijenite parametar pokretanja petlje
...	...
5	a) Upotrijebite pojedinačni uglavljeni sloj petlje bez skočne poruke b) Koristite višestruko ugniježdene petlje u nizu sa ili bez poruke c) Upotrijebite ugniježdene petlje s više od jednog sloja u dubinu uz skočnu poruku

Izvor: Code.org 2017 Godišnji izvještaj, <https://code.org/about/2017>

Za tu matricu označili su svaku od 500 zagonetki na svim tečajevima CS Osnove računalstva kako bi povezali svaku zagonetku s jednim ili više pojmova, a za svaki koncept, ocjenu poteškoća. Na primjer, slagalica prikazana u nastavku (Slika 1.) ima

različite razine težine za različite pojmove. Označeno je: *Sekvenciranje-4*, „Repeat Until“ *petlja-2* i *Uvjeti-2*.

Slika 1. Primjer zagonetke sa raznim težinama zadatka

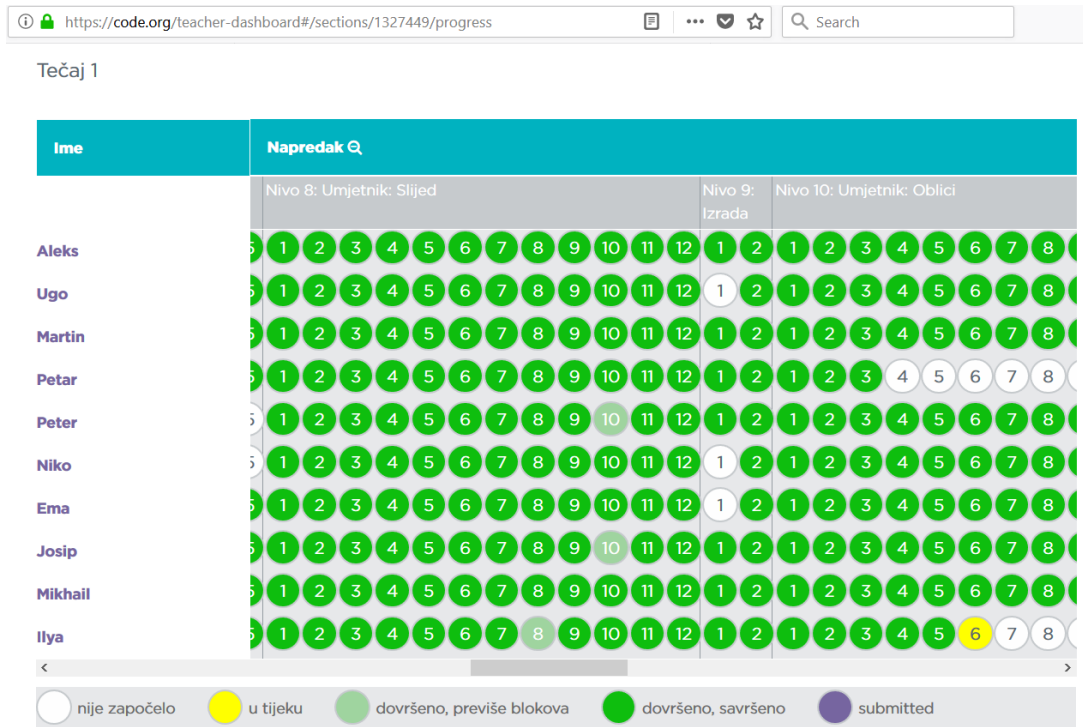


Izvor: Code.org stranica <https://studio.code.org/s/course3/stage/8/puzzle/3>

Mjerenje uspješnosti učenika u Code Studio-u može se objasniti kroz tri čimbenika, točnije odgovora na pitanja:

1. Jesu li ispravno dovršili zagonetku? Na mnogim zagonetkama učenik ne može dovršiti zagonetku bez ispravnog korištenja koncepta. Učenici koji ga ne razumiju će pokušati iznova ili će preskočiti na sljedeći zadatak ostavljajući ga privremeno nepotpunim i označenim svijetlo zelenom bojom a ne tamno zelenom kao što bi to bilo da su složeni blokovi koda bili u potpunosti točni. Samo se računa zagonetka koja je ispravno riješena. Svaki edukator ima mogućnost praćenja napretka svojih učenika u razredu pomoću Upravljačke ploče učitelja (Slika 2.). Pregledom odabranog tečaja učitelj može vidjeti za svakog učenika svaki njegov pokušaj rješavanja zadatka i status koji je dobio taj zadatak. Status zadatka može biti: **nije započeto**; **u tijeku** (što može značiti ili da trenutno rješava ili da je makar pogledao jednom zadatak i odabrao neki drugi zadatak za rješavati), **dovršeno**, **previše blokova** ali nije korišten optimalan broj blokova; **dovršeno**, **savršeno**; te **predano** što je namijenjeno kviz zadacima u kojima se traži da izaberu točan odgovor od nekoliko ponuđenih odgovora.

Slika 2. Autorova upravljačka ploča i napredak učenika



Izvor: autorov profil i napredak učenika za Tečaj 1 na Code.org platformi

2. Jesu li završili slagalicu s optimalnim brojem blokova? Većina zagonetki na Code Studio pokazuje "optimalni" broj blokova - očekivani broj linija koda u željenom rješenju. Code Studio smatra kako je korištenje optimalnog broja blokova pokazatelj "stručnosti" jer u mnogim slučajevima optimalni broj blokova nije moguće ostvariti bez primjene željenog koncepta - to je osobito važno za zagonetke s petljama. Na primjer, možete proći zadatke slaganja blokova koda bez petlji tako da ručno ponavljate redove koda koliko je potrebno, ali samo učenici koji koriste petlje mogu proći s optimalnim brojem blokova. Na slijedećoj slici (Slika 3.) vidi se statistika učenika koji su rješavali Tečaj 1 gdje se može uzeti u razmatranje učenik pod korisničkim imenom Martin i Ilya. Ilya ima ukupno 754 linija koda što je najbliže rezultatu koji je postigao Martin sa 792 linije koda. Razlika između Ilye i Martina je u tome što Martin ima i najviši završen nivo 95 dok Ilya ima 58 nivo, tj. razinu. To što učenik ima veći broj linija koda ne znači da je došao do optimalnog rješenja i dokazao da je savladao koncepte. Potrebno je uzeti u obzir broj linija kodova ali i nivo koji je postignut.

Slika 3. Statistika učenika koji su rješavali Tečaj 1

Ime	Završeni nivoi	Linije koda
Aleks	78	583
Ugo	66	445
Martin	95	792
Petar	51	342
Peter	50	672
Niko	62	242
Ema	69	387
Josip	74	457
Mikhail	61	547
Ilya	58	754

Izvor: autorov Code.org profil i prikaz statistike učenika u Tečaj 1 na Code.org platformi

3. Jesu li oni dovršili zagonetku bez upotrebe savjeta? Mnoge zagonetke nude jedan ili više savjeta koje učenik može odabrati klikom na tipku "savjet". Studio Code uzima riješeni zadatak kao dokaz sposobnosti učenika ako ga učenik riješi bez ikakvih savjeta. Ti se savjeti drastično razlikuju u količini pomoći koju daju, od blagog gurkanja u pravom smjeru do nečega što u biti ne daje točan odgovor ili ga „skriva“. I savjeti također imaju svoje napredovanje. Važno je napomenuti da Code.org sustav daje neke prilagođene poruke o pogrešci kada se pokušaju pogrešna rješenja. Prilagođene poruke o pogrešci su oblik nagovještaja. Dakle, da bi ga održali jednostavnim, samo su računali dovršavanje slagalice prema znanju ako učenik nije tražio nikakve savjete. Postoji mnogo dodatnih čimbenika koji mogu biti pokazatelji znanja kao što su duljina vremena za rješavanje zagonetke, broj pokušaja ili određenih rješenja. Međutim, kao prvi pokušaj mjerenja korišteni su samo indikatori koji su nedvosmisleno pokazali uspjeh. Veliki broj pokušaja rješavanja zagonetke, na primjer, može ukazivati na polaznike koji polako grade svoj kod i testiranje dok idu zajedno (uobičajena najbolja praksa) ili bi mogao biti učenik koji se bori da bi pronašao točan odgovor.

Određivanje stručnosti učenika: Koncepti + Težina

Određivanje "stručnosti" pomoću Code.org sustava označavanja djeluje pomalo proizvoljno i nijansiran. No, kao metriku visoke razine htjeli su biti u mogućnosti reći da je X broj studenata koji koriste njihovu platformu pokazao "osnovnu sposobnost kodiranja". To je zahtijevalo da definiraju odluke pri određivanju stručnosti odgovorom na slijedeća pitanja:

- 1. Koliko različitih zagonetki učenik mora ispravno dovršiti?**
- 2. Koliko različitih koncepata mora ispravno dovršiti?**
- 3. Na kojoj težinskoj razini trebaju biti ta rješenja?**

Odluke pri određivanju stručnosti odgovaraju na prethodno postavljena pitanja.

Koliko zagonetki? Tri.

Za svaku zagonetku koju učenik ispravno dovršava, s optimalnim brojem blokova i bez savjeta, učenik dobi bod za savladani koncept ili težinsku razinu prilikom rješavanja koncepta. Na primjer: ako je određena zagonetka imala oznake za: Algoritmi - 2 i Petlja ponavljanje do - 4 onda bi student dobivao 1 bod za algoritme na razini težine 2, i bod za ponavljanje petlji u razini težine 4.

Važno je napomenuti da rješavanje zagonetki s većom težinskom razinom također računa kod priznavanja sposobnosti naučenog učenika i za nižu težinsku razinu. Dakle, na primjer, učenik koji završi zagonetku s oznakom „repeat“ petlja - 4 dobiva bodove prema stručnosti za težinsku razinu 1-3 također. To se može opisati ovako: Učenik je pokazao stručnost u konceptu X, poteškoće Y kada je riješio tri zagonetke označene konceptom X i težinom Y ili više.

Važno je napomenuti i da su parovi poput algoritama i težina 3 prisutni u stotinama zagonetki, a također su i vrlo jednostavni - možete pokazati vještinu unutar samo nekoliko pokušaja zagonetke i imati mnogo prilika da to ponovno pokažete. Drugi parovi, poput for-petlja u poteškoćama 5, ne primjenjuju se na mnogo zagonetki, a i te zagonetke često imaju prilično zahtjevne izazove.

Koliko koncepata? Tri.

Osnovno računalno kodiranje zahtijeva od učenika da nauče više pojmova koji međusobno djeluju i utječu jedni na druge, ali nema uspostavljenog "ispravnog" reda u kojem ih treba naučiti. To znači da bi se moglo doći do uvjeta prije petlje i obrnuto. Stoga su obratili pozornost na učenike koji su pokazali sposobnost za određeni broj

različitih pojmova. Kao početnu procjenu odabrali smo pogledati učenike koji su pokazali znanje u tri (3) različita područja koncepata.

Koliko teško? Razina težine 3.

Na kraju je ustanovljeno da bi se od 1-5 ocjenjivanje moglo postaviti u tri težinska područja.

Tablica 5. Prikaz temeljna područja težine i opis težinske razine

Težina 1 – 2	Težina 3	Težina 4-5
Težina 1 i 2 imaju tendenciju mjerenja i uvođenja mjera ili osnovne izloženosti konceptu ili ideji. Učenik ne mora razumjeti koncepte za rješavanje zagonetki na ovim razinama. Iako mogu upotrebljavati petlje, uvjete i sl., slaganje kodnih blokova je jednostavno ili se od učenika traži da ih presloži na ispravan način.	Težina 3 obično je zagonetka u kojoj učenici moraju primijeniti petlje, uvjetne ili druge pojmove, dok rješavaju problem. Upute za ove zagonetke također imaju tendenciju da budu manje eksplicitne ili bez puno opisa, a više nalik otvorenom pitanje-izazovu: "Možete li to riješiti?"	Težina 4-5 obično su zagonetke koje su jedna ili kombinacija otvorenih uputa, duljih rješenja, složenije primjene ideja ili čak zahtijevaju nove pristupe u rješavanju novih situacija.

Izvor: Code.org 2017 Godišnji izvještaj, <https://code.org/about/2017> (prijevod autor)

Budući da je težina 3 bila prva u kojoj su učenici imali priliku rješavati probleme, a ne jednostavno slijediti upute, odabran je stupanj težine 3 kao donja granična točka za dokazivanje "osnovne vještine".

U slijedećem poglavlju pokazati ću na koji način se koristila web aplikacija prilikom održavanja nastave Digitalne akademije. S obzirom da je takva nastava koncipirana kao projektna nastava u nastavku slijedi objašnjenje i kratak povijesni pregled projektne nastave kao suvremenog oblika nastavnog sata.

PROJEKTNNA NASTAVA

Projektna nastava je didaktički sustav nastave koji se odvija po projektu koji omogućava samo-organizirajuću i samo-odgovornu nastavu u kojoj se pojedinačni učenikov rad nadovezuje na pomoć suučenika i nastavnika. U objašnjenju projektne nastave, kao oblika nastavne organizacije, moguće je shvatiti kao primjerenu samoorganizaciju. Prije svega zbog osamostaljivanja učenika u učenju. No ipak u projektnoj nastavi ne može se očekivati idealna i potpuna samostalnost učenika. Projekt predstavlja zajednički pokušaj nastavnika i učenika da učenje i rad povežu tako da postavljeni zadatak zajednički obrade i da rad dovedu do rezultata-produkta (Jurčić, 2007.).

Ono što je osobito važno, u pogledu razvoja učenikovih socijalnih kompetencija, je to da učenici u projektnoj nastavi razvijaju odgovornost za zajednička postignuća, sposobnost ocjene i objektivne procjene nekog rada, uče konstruktivnu kritičnost, uče prepoznati pozitivne učinke drugih, uče prihvatiti objektivne procjene drugih i slično. (Buljubašić-Kuzmanović, 2007.) U projektnoj nastavi teži se uravnoteženom odnosu između misaonog i tjelesnog rada koji može biti na razini blok sata, projektnog dana, projektnog tjedna ili na razini većeg projektnog razdoblja što ovisi o samom projektu. Na kraju ovog poglavlja biti će prikazani primjeri aktivnosti provedene projektne nastave informatike koja se izvodila tijekom školske godine 2017./2018. u sklopu Algebrine Digitalne akademije u OŠ Vidikovac. U tim primjerima izdvojeni su dijelovi kada se koristila web aplikacija Code.org za učenje osnova računalstva. Algebra je za potrebe Digitalne akademije kreirala nastavni plan i program za digitalnu i informatičku obuku učenika osnovnih škola temeljen na projektnoj nastavi. Za svaki je razred kreirano po 7 projekata u trajanju od 8 do 12 školskih sati. Kao informatički instruktor-edukator na Digitalnoj akademiji provodim projektnu nastavu po planu i programu Algebre što će biti i prikazano u navedenim primjerima.

Povijesni pregled projektne nastave

Projektna nastava se po prvi puta koristi potkraj 19. stoljeća zahvaljujući američkom pedagogu James Deweyu (1859. – 1952). Dewey je osnovao vlastitu eksperimentalnu školu, gdje je koristio svoje pedagoške koncepte. Dewey je tijekom te eksperimentalne škole zaključio da dječje osobne interese ne može zadovoljiti samo škola, gdje su u prvom planu nastavni predmeti i razredne grupe. Smatrao je da dijete ima četiri osnovna interesa: interes za komuniciranje s ljudima, interes za istraživanje, interes za rad i interes za umjetničko izražavanje. (Lučić i Matijević, 2004., str. 58)

Stoga je u svom radu koristio nastavne projekte umjesto nastavnih predmeta. Na temelju toga shvaćanja, Dewey je u odgoju djece koristio aktivnosti koje uvažavaju dječji interes. „Njegov je pedagoški koncept realiziran u formi konkretnog rada djece u školskim radionicama i laboratorijima, te u školskom vrtu, voćnjaku, polju i drugdje u prirodi“ (Bognar i Matijević, 2005., str. 60)

Na temelju navedenog rada John Deweya, W. H. Kilpatrick je razvio i dalje nadogradio takvu vrstu odgoja i školsku praksu. Izdao je knjigu „*The Project Method*“ u kojoj je izložio koncept nastavnog rada po projektima. U svojoj knjizi naveo je četiri vrste koncepta rada na projektima:

- 1) Projekti oblikovanja (modeliranja, igre)
- 2) Projekti estetskog doživljavanja
- 3) Projekti rješavanja problema
- 4) Projekti uvježbavanja vještina

Uz dogovor s nastavnicima učenici rade na projektu podijeljeni u grupe, parove ili individualno. (Bognar i Matijević, 2005., str. 60)

Sam rad na projektu podijelio je u šest osnovnih faza:

1. Zajedničko postavljanje problema
2. Iznošenje pretpostavki (hipoteza) za rješavanje problema
3. Izrada plana za rješavanje problema
4. Izvođenje rada prema definiranom planu
5. Izvođenje zaključaka
6. Primjena zaključaka u praktičnom životu

Iako je uvedena u školsko obrazovanje tijekom 30.-ih godina prošlog stoljeća, projektna nastava je dobila značajniji napredak u školskom obrazovanju širenjem timske nastave u svijetu, sredinom 60.-ih godina prošloga stoljeća. Tada se u Velikoj

Britaniji počela primjenjivati timska nastava u eksperimentalnim školskim programima. 1966. godine počela se primjenjivati i u SR Njemačkoj, unatoč brojnim kritikama koje nisu odobravale njeno korištenje u nastavi. U početku je bila izvođena putem blok satova i grupne nastave.

Primjeri provedenih aktivnosti sa učenicima 1. razreda

Aktivnost 1 – Moja jutarnja rutina – usvajanje pojma algoritam

Učiteljica započinje razgovor kojim učenike potiče da navode koje aktivnosti obavljaju prije jutarnjeg polaska u školu. Uočavamo kako je redoslijed bitan i kako je važno da se neke aktivnosti obave prije ostalih. Zato učiteljica organizira igru: učenici sjede u krugu. Jedan učenik navodi jednu aktivnost (npr. perem zube). Učenik pored njega ponavlja aktivnost koju je naveo prvi učenik i dodaje još jednu – pazeći na redoslijed (npr. budim se, perem zube). Igra je gotova kada svi učenici dodaju po jednu aktivnost i sve ponove pravilnim redoslijedom ili kada jedan učenik pogriješi.

Učiteljica objašnjava kako svaka od navedenih aktivnosti opet ima niz koraka koji treba napraviti kako bi se aktivnost uspješno obavila. Odaberemo primjer aktivnosti: pranje zuba. Učenici se dogovaraju u paru, a zatim svaki par navodi koje sve korake treba izvršiti kako bi se uspješno oprali zubi (npr. uzimamo četkicu, navlažimo četkicu, stavljamo pastu...). Osobito su pohvaljeni parovi koji nabroje najviše koraka pravilnim redoslijedom. Potrebno je zaključiti kako nije lako prisjetiti se svih koraka jer ljudi neke korake rade automatski, da o njima ni ne razmišljaju. Mi moramo uvježbati uočavanje svih, pa i najsitnijih koraka, jer ćemo samo tako biti uspješni u programiranju – pisanju uputa računalu što i kako činiti. A računalima trebamo dati baš sve, pa i najsitnije upute!

Aktivnost 2 – Kako prijeći cestu kao robot?

Učiteljica potakne učenike da se prisjete što sve treba učiniti kako bi se sigurno prešla ulicu kada nema semafora (doći do pješačkog prijelaza, pogledati lijevo-desno, pričekati da se sva vozila zaustave, žurno prijeći preko pješačkog prijelaza).

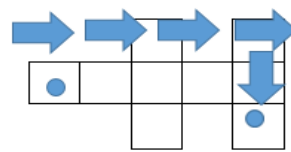
Učiteljica pozove učenike na igru: Ljudski robot želi prijeći cestu. Na podu učionice kredom nacrtamo „rubove“ kolnika i nekoliko debljih crta kao pješački prijelaz. Učenicima objasnimo kako će učiteljica biti robot koji 1 učenik treba upravljati

jednostavnim naredbama (naprijed, lijevo, desno, stani). Učiteljica sluša naredbe doslovno, kako bi osvijestila učenicima važnost davanja preciznih i detaljnih uputa (npr. jedan korak desno, a ne: idi desno). Nakon demonstracije učiteljice i jednog učenika, igru igraju parovi učenika. Dva učenika sudjeluju u igri dok drugi sjede oko „ceste“ i pamte što je bilo dobro, a što loše. Na kraju igre učenici komentiraju dobre/loše upute koje su učenici davali, te pohvaljujemo najpreciznije „upravljače robota“.

Aktivnost 3 – Igra Školice na ploči

Učiteljica najavljuje još jednu igru upravljanja kratkim i preciznim naredbama. Učenici prepoznaju polja školice nacrtane na ploči i 2 magneteta koja se na njoj nalaze. Učenici će igrati u malim grupama: jedan će učenik razmjestiti magnet na poljima Školice i odrediti koji magnet treba stići do svog prijatelja. Ostali slažu strelice koje predstavljaju upute za kretanje. Učenik koji je razmjestio magnet pokušava slijediti upute strelicama kao što je to prikazano na slici 4.

Slika 4. Igra Školice na ploči



Izvor: Projekt 2 – Ljute ptice – namijenjeno učenicima 1. razreda (Algebra, Digitalna akademija)

Aktivnost 4 – Povuci – spusti tehnika slaganja blokova

Učiteljica najavljuje da će svaki učenik na svom računalu vježbati pravilno služenje mišem, kao i tehniku “povuci-spusti”. Učiteljica upućuje učenike da otvore Razrednu e-mapu (engl. *Portfolio*) i da pronađu Word dokument na kojemu piše njihovo ime. Prepisuju Internet adresu u svoj preglednik.

Slijedi nastavak igre sa likovima iz igrice „*Angry Birds*“ u Školici. Grupe sa prošlog susreta izmjenjuju se na ploči za Školicu. Jedan učenik iz grupe postavlja likove u „mrežu“ za Školicu koja je nacrtana na ploči, drugi učenik slaže strelice pored „mreže“ – po kojima bi ptica trebala uloviti svinju, a treći učenik pokreće lik ptice prema tim uputama - strelicama.

Nakon što sve grupe probaju slagati upute i eventualno ispraviti uočene pogreške, učenici iz drugih grupa koji su gledali, komentiraju dobre/loše poteze i učiteljica ih potiče da pohvaljuju uočavanje pogrešaka, te ispravljanje istih.

Aktivnost 5 – Ljuta ptica u programiranju

Učiteljica demonstrira otvaranje Code.org / Tečaj 1 / Nivo 4 i način rada kako je i prikazano na slijedećoj slici.

Slika 5. Upoznavanje sa sučeljem Code.org tečaja



Izvor: Projekt 2 – Ljute ptice – za učenike 1. razreda (Algebra, Digitalna akademija)

Tehnikom „povuci-spusti“ složiti blokove kako da ptica dođe do svinje. Učiteljica potiče učenike pitanjima:

- Koju strelicu/blok trebamo koristiti? (lijevo)
- Koliko je puta trebamo koristiti? (dva puta)

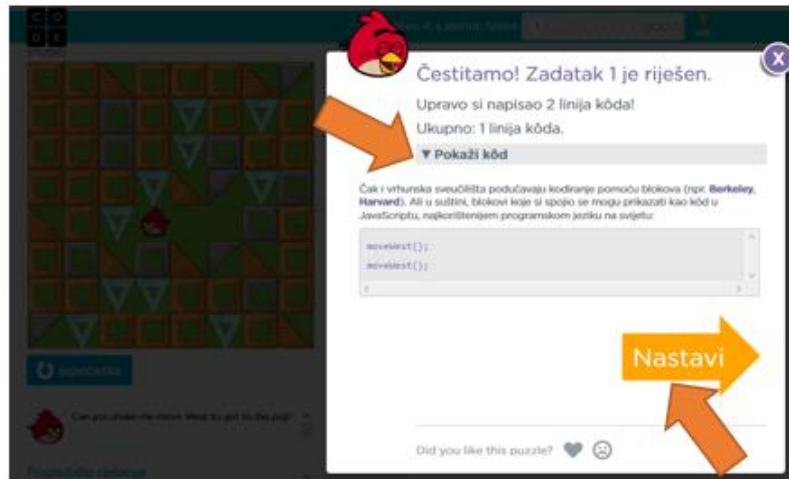
Nakon što je učenik provjerio svoj uradak, isproba ga klikom na gumb Pokreni.

Ukoliko smo napravili pogrešku, program nas potiče da pokušamo ponovno.

Ukoliko smo točno riješili zadatak, program nam daje mogućnost pogledati kako je računalo razumjelo slaganje blokova.

Učiteljica potiče učenike da nakon svake igre pogledaju kod te objašnjava učenicima kako računalo ne možemo ljudskim riječima niti blokovima objasniti kako da učini da ptica dođe do svinje. Složene blokove računalo pretvara u naredbe na jeziku koji on razumije – za svaki blok piše jedan red koda. Ovi redovi se zovu **slijed naredbi** koje računalo razumije i može izvršiti a sve zajedno se zove **računalni program** napisan **programskim jezikom** (možda će učenike i zanimati kako se taj jezik zove – ovo je najkorišteniji programski jezik na svijetu za razvoj web stranica, a zove se *JavaScript*).

Slika 6 Upute kako pokazati Javascript kod u Code.org



Izvor.: Projekt 2 – Ljute ptice – namijenjeno učenicima 1. razreda (Algebra, Digitalna akademija)

Klikom na tipku Nastavi prelazimo na iduću igru slaganja. Učenici u ovom nivou mogu ukupno odigrati 15 igara.

Napomena: Za strelice/blokove koristiti instrukcije: lijevo, desno, gore, dolje jer učenici strane svijeta uče tek u 3. razredu. Učiteljica ne treba inzistirati na tome da svi učenici za vrijeme trajanja ove aktivnosti odigraju svih 15 igara. Učenicima koji prije završe ponuditi Nivo 10 – umjetnik – oblici.

Aktivnost 6 - Predstavljam svoj rad

Učiteljica pitanjima potiče učenike da predstave svoj rad u Code.org Nivo 4:

- Koliko si igara odigrao/la?
- Koja ti je igra bila najlakša/najzahtjevnija?
- Što misliš o mogućnosti da ispraviš pogreške u svom radu?
- Što misliš o programskom jeziku *Javascript*?

Ponoviti osnovne pojmove i sadržaje iz cjeline Programiranje, i istaknuti da je cilj ovakvih aktivnosti njegovati upornost i dosljednost u ispravljanju pogrešaka nastalih prilikom sastavljanja jednostavnih programa. Učiteljica pitanjima potiče učenike da se prisjete što smo do sada naučili o programiranju:

- Razumiju li računala naš jezik? (ne)
- Kako zovemo jezike koje računala razumiju? (programski jezici)

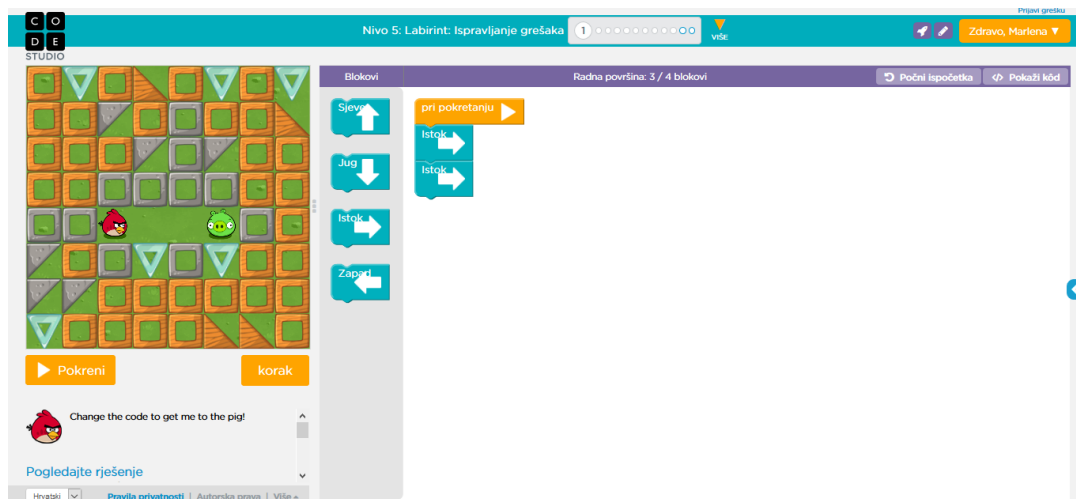
- Što smo mi koristili za sporazumijevanje sa računalom? (strelice, blokove)
- Može li računalo ispraviti naše pogreške?
- Zašto je važno ispravljati pogreške? (Računalo ne može samo ispravljati pogreške. Dok ne ispravimo pogrešku računalo ne može razumjeti što mi želimo postići.)

Najaviti nastavak igre sa ljutom pticom i programiranjem. Osobitu ćemo pažnju posvetiti njegovanju upornosti i marljivom ispravljanju pogrešaka!

Aktivnost 7 – Ljuta ptica u programiranju

Učiteljica demonstrira otvaranje Code.org / Tečaj 1 / Nivo 5 i način rada što je vidljivo nas slijedećoj slici.

Slika 7. Zadani kod (složene blokove) koje treba ispraviti



Izvor: Projekt 2 – Ljute ptice – namijenjeno učenicima 1. razreda (Algebra, Digitalna akademija)

Kod svake igre u Nivou 5 imamo već zadani kod (složene blokove) koji treba promijeniti (pronaći i ispraviti pogreške) kako bi ptica došla do svinje. Ako želimo neki od blokova izbrisati, jednostavno ga kliknemo i pomaknemo ulijevo. Možemo pokušavati i ispravljati koliko god puta želimo. Ukoliko smo točno riješili zadatak, program nam daje mogućnost pogledati kako je računalo razumjelo slaganje blokova a učiteljica potiče učenike da nakon svake igre pogledaju linije koda u *Javascriptu*. Svaki učenik rješava na svom računalu i prema svojim mogućnostima. Sve učenike učiteljica hrabri, potiče i hvali da ustraju do 12. igre – kraja Nivoa 5 (premda to ne treba biti uvjet).

Aktivnost 8 – Predstavljam svoj rad

Učiteljica pitanjima potiče učenike da predstave svoj rad u Code.org Nivo 5:

- Koliko si igara odigrao/la?
- Koja ti je igra bila najlakša/najzahtjevnija?
- Što misliš o mogućnosti da ispraviš pogreške u svom radu?
- Je li lakše slagati sve od početka ili tražiti pogreške u tuđem radu?

Učiteljica objašnjava učenicima kako je vještina uočavanja pogrešaka u gotovom uratku jedna od najvažnijih i najcjenjenijih vještina u radu, te da moramo biti uporni jer se upornost uvijek isplati!

Aktivnost 9 – Programeri kao umjetnici

Učiteljica najavljuje nastavak rada na aplikaciji Code.org. Za razliku od prethodne aktivnosti u kojoj su u Word dokument umetali pune crte i tako dovršavali crtež, sada će trebati slagati programske blokove pravilnim redoslijedom i tako stvarati programe koji će umjesto njih dovršavati crteže. Učenici otvaraju e-portfolio i pronalaze Word dokument sa svojim imenom. Prepisuju Internet adresu u svoj preglednik.

Učenici će odigrati 12 jednostavnih igara sa Tečaj 1; Nivo 8. Do pete igre trebaju slagati blokove i samo paziti na smjer pružanja blijeđe crte koju trebaju podebljati.

Napomena: Učenici tek u 3. razredu iz Prirode i društva uče strane svijeta. Kod slaganja blokova trebaju se fokusirati na smjer koji pokazuje strelica na svakom programskom bloku, a ne na stranu svijeta koja piše ispod strelice.

Od 6. do 11. igre pojavljuju se i blokovi kojima preskačemo dio crteža, odnosno pisaljka nastavlja kretanje ali ne ostavlja trag na papiru. Takve blokove učenici slažu na isti način kao i blokove za crtanje. U 12. igri učenici trebaju pažljivo promotriti složene algoritme i odlučiti se za onaj koji pokazuje redoslijed naredbi koji će omogućiti čovječuljku sa olovkom (lijevo) da uspješno nacrtava kvadrat kao što je prikazano na slici broj 8.

Slika 8. Kviz pitanje višestruki izbor



Slika 7.: Projekt 2 – Ljute ptice – namijenjeno učenicima 1. razreda (Algebra, Digitalna akademija)

Točnost svog odgovora provjeravaju klikom na gumb Pošalji. Točan je prvi ponuđeni odgovor. Za slijedeću aktivnost učiteljica treba poticati učenike da verbaliziraju niz uputa za kretanje „ljudskog robota“. Učenici trebaju prepoznati glavni cilj zadatka i odrediti koje postupke treba primijeniti za njegovo uspješno rješavanje. Tijekom aktivnosti bi trebali upoznati različite programske blokove, njihove mogućnosti i primjenu u programiranju.

Aktivnost 10 – „Ljudski robot“ crta kvadrat

Učiteljica prisjeća učenike posljednje aktivnosti koju smo odigrali na prošlom susretu: kako nacrtati kvadrat? Učenici odigraju igru „ljudskog robota“: jednog učenika postavimo ispred ploče, dodamo mu u ruku kedu i stavimo povez preko očiju, dok mu drugi učenik daje upute kako da nacrtava kvadrat na ploči. Važno je napomenuti da sada nemamo programske blokove već „ljudski robot“, stoga moramo uvesti i naredbu „stop“ kako bi učenik znao kada prestati vući crtu. Tako bi algoritam usmenih naredbi za crtanje kvadrata na ploči mogao glasiti ovako:

- Ispruži ruku sa kredom
- Kredu pritisni na ploču
- Vuci liniju desno
- Stop
- vuci liniju gore
- Stop
- Vuci liniju lijevo

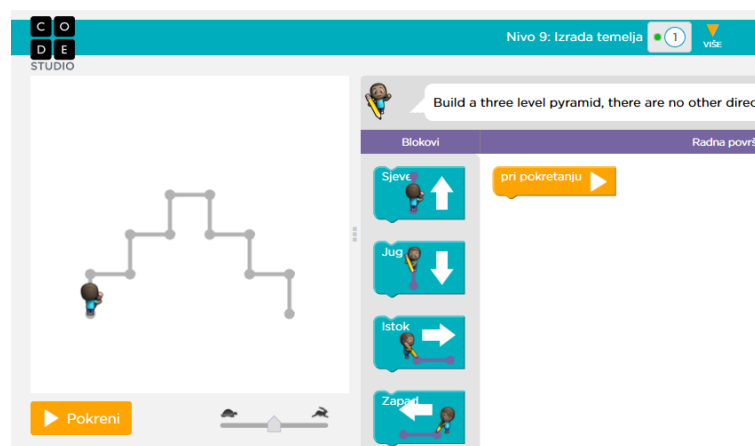
- Stop
- Vuci liniju dolje
- Stop

Nakon što par učenika odigra igru „ljudskog robota“, razredu iznesu svoja iskustva: kako im je bilo slušati/davati naredbe, što im je bilo najzahtjevnije, a što najzabavnije, najlakše i sl.

Aktivnost 11 – Izgradi piramidu

Učiteljica najavljuje nastavak rada na aplikaciji Code.org: Tečaj 1, Nivo 9, zadatak 1. Koristeći samo 4 vrste programskih blokova za 4 osnovna smjera učenici će trebati izgraditi piramidu sa 3 nivoa. Svaki učenik otvara vlastitu mapu *e-portfolio* i pronalazi Word dokument sa svojim imenom. Prepisuje Internet adresu u svoj preglednik i započinje graditi piramidu prikazanu na slijedećoj slici.

Slika 9. Code.org: Tečaj 1, Nivo 9, zadatak 1.



Izvor: Projekt 2 – Ljute ptice – namijenjeno učenicima 1. razreda (Algebra, Digitalna akademija)

Na kraju učenici komentiraju svoj uradak: što im je bilo najteže i kako su riješili probleme. Stvaraju snimak zaslona sa uspješno riješenim zadatkom i spremaju ga u *e-portfolio*.

Aktivnost 12 – Crtamo likove

Učenici nastavljaju rad na prvih 9 zadataka u Code.org, Tečaj 1, Nivo 10. Sada će spajati točke kako bi crtali kvadrate i pravokutnike koji upotpunjuju gotove crteže. U 6. zadatku biti će potrebno podebljavati već nacrtane stranice kako bi se moglo nastaviti sa crtanjem. U 8. zadatku koristit ćemo blok za preskok, dok u 9. zadatku tehnikom

povuci-spusti treba sličice iz desnog stupca pravilno spojiti sa sličicama sličnih likova iz lijevog stupca. Za slijedeću aktivnost ciljevi bi bili uvježbati tehniku slaganja programskih blokova i ispravljanja pogrešaka u programiranju, ponoviti i praktično primijeniti usvojene geometrijske sadržaje, pa stvoriti vlastiti digitalni uradak.

Aktivnost 13 – Ja kao umjetnik

Učenici pomoću poveznice u Wordu ponovno pristupaju Code.org, 10. nivo, 10. zadatak. U ovom zadatku imaju priliku primijeniti sve što su do sada naučili o slaganju programskih blokova, te nacrtati što god žele: osim blokova koje su već upoznali u ovom zadatku uvedeno je i još nekoliko novih blokova za crtanje kosina (označene slovima za sporedne strane svijeta). Na dnu izbornika programskih blokova nalazi se i blok za određivanje boje linije koji učenici trebaju umetnuti prije blokova za crtanje. Učiteljica potiče učenike da prvo razmisle o osnovnom motivu koji žele nacrtati, a da zatim u svoj crtež pokušaju uvrstiti što više detalja – likova u nizu kako bi crtež bio što bogatiji i kako bi u potpunosti i kvalitetno iskoristili svo predviđeno vrijeme za ovu aktivnost.

Aktivnost 14 - Pas i mačka razgovaraju o Pokemonima

Učenici će odigrati i 6. igru 16. nivoa u Code.org. Ovdje mogu iskoristiti sve blokove sa kojima su se upoznali u prvih pet igara. Neka njihovi likovi porazgovaraju o likovima iz Pokemon crtanog filma. Učenici će prvo isprobati koje mogućnosti daje koji blok, što će im dati ideju za stvaranje priče. U tablici 6. imamo prikaz likova, pozadina, brzine kretanja i raspoloženja koja možemo koristiti prilikom kreiranja vlastite strip priče.

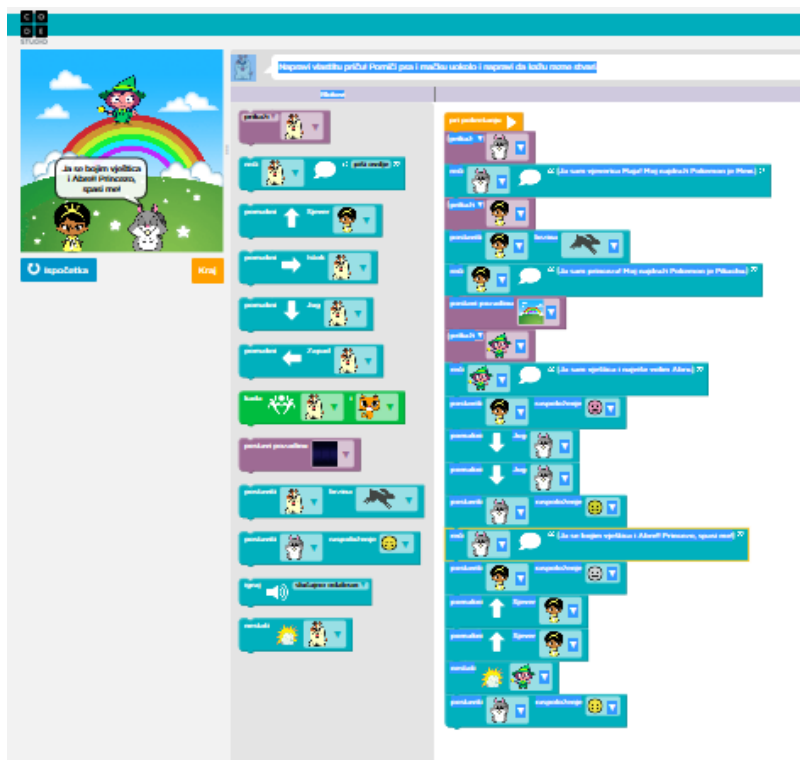
Tablica 6. Popis likova, pozadina, brzine kretanja i raspoloženja

Likovi:	Pozadine:	Brzine kretanja:	Raspoloženja:
			

Izvor: Projekt 5 – U svijetu Pokemona – programiranje strip priče, namijenjeno učenicima 1. razreda (Algebra, Digitalna akademija)

Učiteljica na razrednom računalu započne slagati i razvijati vlastitu priču. Dobro je analizirati takvo rješenje i uočiti slabe strane i mjesta gdje se priča mogla popraviti (npr. princeza je mogla nešto reći kada je tjerala vješticu). Poticati učenike da budu kreativniji i da se ohrabre te krenu u stvaranje vlastite priče i truditi se napraviti bolju priču od pokazane. Slijedi primjer jednog gotovog rada strip priče učenika 1. razreda polaznika Digitalne akademije.

Slika 10. Kodiranje strip priče, rad učenika 1.razreda



Izvor: Projekt 5 – U svijetu Pokemona –rad učenika 1. razreda (Algebra, Digitalna akademija)

Primjeri provedenih aktivnosti sa učenicima 3. razreda

Aktivnost 1 – Algoritmi

Prisjećamo se otprije naučenog gradiva kako bismo u novoj priči – novom projektu bili što uspješniji. Učiteljica najavljuje da ćemo se i mi baviti PROGRAMIRANJEM – pisati ćemo jednostavne upute računalu što i kako treba učiniti. Danas ćemo ponoviti i značenje drugih pojmova koji su vrlo važni za uspješno programiranje:

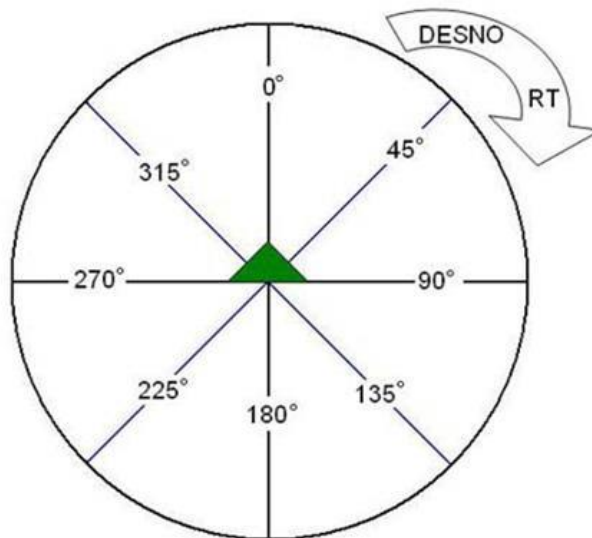
ALGORITAM je niz koraka koji treba slijediti kako bi se izvršio zadatak. Što smo sve trebali jutros napraviti prije nego smo krenuli u školu? Učitelj-ica potiče učenike da navedu još primjera iz svakodnevnog života u kojima trebamo slijediti niz koraka potrebnih za izvršenje zadataka: pozovimo nekoga mobitelom, napravimo sendvič... Učenici u svojem *e-portfolio* (prezentaciji) stvaraju novi slajd (ili više njih) oblika *Naslov i sadržaj*, stavljaju naslov „Robot izlazi iz učionice“ i pišu algoritam za ljudskog robota. Uputama ga trebaju odvesti od svojeg stola do izlaza iz učionice, uključivši i izlazak iz učionice.

Učenici u parovima testiraju svoje algoritme ljudskom robotu, jedan učenik čita svoj algoritam, a drugi glumi robota koji radi samo ono što mu se kaže.

Zajednički promatramo upravljanje robotima, pohvalimo dobra rješenja i smišljamo kako ispraviti pogreške. Istaknuti naredbe za okretanje – što znači okrenuti se desno? Za koliko desno?

Uvodimo pojam kuta i stupnjeva:

Slika 11. Grafički prikaz stupnjeva



Izvor: Projekt 1 – Početak nove priče – učenici 3. razreda (Algebra, Digitalna akademija)

- Desno = 90°
- Na suprotnu stranu = 180°
- Puni krug = 360°
- Lijevo za 90° = Desno za 270°
- Lijevo 180° = Desno 180°

Kod kretanja naprijed i natrag brojali smo korake u zadanom pravci, međutim kod kretanja lijevo i desno moramo najprije shvatiti kako se odvija kretanje u krugu:

1. Zamislimo da, ako želimo napraviti cijeli krug oko sebe kako bismo stigli u položaj u koji smo krenuli, trebamo 360 stupnjeva. Učiteljica napiše naredbu na ploči i potiče učenike da se dignu sa svojih mjesta i naprave LT 360 (zavrte se oko sebe ulijevo i stignu u početni položaj). Isto napravimo i za RT 360.

2. Ako želimo gledati iza sebe trebamo napraviti pola kruga, a pola je 180 stupnjeva. Učiteljica napiše naredbu na ploči i potiče učenike da se dignu i naprave LT 180 (zavrte se oko sebe ulijevo kako bi gledali iza sebe). Isto napravimo i za RT 180

3. Učenici rašire ruke. Što mislite, koliko bi stupnjeva bilo potrebno za okret ulijevo odnosno udesno – kamo pokazuje jedna od raširenih ruku? (90). Učiteljica zadaje: LT 90; RT 90

Prisjećamo se kako računala ne razumiju ni jedan od jezika kojima se sporazumijevaju ljudi, već da im mi moramo prevesti naše zahtjeve na jezik koji oni razumiju, a to su PROGRAMSKI JEZICI. Iskusni programeri znaju pisati programe na mnogim programskim jezicima (možda će učenici znati kako se neki od njih zovu: *Java*, *JavaScript*, *C#*, *C++*, *Visual Basic*...), a mi, „mali programeri“ služiti ćemo se jednostavnim naredbama i programskim blokovima – likovima koji predstavljaju naredbe koje razumiju i računala i mi.

Aktivnost 2 – Programiramo ili se igramo?

Učiteljica će podijeliti učenicima kartice za učenike s informacijama za prijavu u razred otvoren u studio.code.org – Tečaj 2. Učenici otvaraju stranicu <https://studio.code.org/s/course2> **Tečaj 2**, rješavaju redom nivoe 1. - 4. Uz rješavanje im učiteljica napiše naredbe na ploči na hrvatskom i engleskom.

- *Move forward* = idi naprijed
- *Move backward* = pomakni unazad
- *By* = za
- *Pixels* = piksela, koraka
- *Degrees* = stupnjevi
- *Turn* = okreni
- *Right* = udesno
- *Left* = ulijevo

Učiteljica pomaže učenicima pri rješavanju, uz davanje savjeta, ali ne i rješenja. Ukoliko se učenici dobro snalaze u programiranju mogu sami birati lekcije koje prolaze. Nakon što su došli do točnog rješenja, blokove s kodovima i svoja zapažanja bilježe u *e-portfolio*. Za svaki zadatak stavljaju novi slajd u prezentaciju, a sliku rješenja kopiraju sa zaslona (Umetanje > Slika zaslona > Zasloni isječak u PowerPointu). Napredak učenika učiteljica prati unutar *Učiteljeve upravljačke ploče*. Neki učenici će s lakoćom rješavati zadatke i takve učenike potičite da rješavaju nivoe do kraja. Također, te učenike treba potaknuti da pomognu onim učenicima kojima baš i ne ide. Učiteljica potiče učenike da pomažu jedni drugima, a ne da govore rješenja.

Aktivnost 3 – Zapetljancije

Ukoliko su učenici početnici, onda im je ovo prvo upoznavanje s petljom. Za napredne učenike može se uvođenje petlje skratiti, a produljiti sljedeća aktivnost. Učiteljica razgovara s učenicima što znači petlja, zapetljati i raspetljati. Zajednički pogledaju video na code.org Tečaj 2 Nivo 5 <https://studio.code.org/s/course2/stage/5/puzzle/1> i nakon toga smišljamo svoju plesnu koreografiju s ponavljanjem – svaki par učenika treba osmisliti jedan korak, a nakon toga izvlače brojeve koliko puta će se taj korak ponavljati. Učenici izvode dogovorenu koreografiju i uz nju govore što je petlja. Učiteljica snima ovu aktivnost u video obliku.

Aktivnost 4 – Petlje

Učenici stvaraju novi slajd u svojem *e-portfolio* i na njemu zapisuju što je petlja. Zajednički kratko ponavljamo: Što su to petlje u programiranju i kada ih koristimo? **PETLJA** nam je potrebna kada se dijelovi programa moraju ponoviti unaprijed zadani broj puta ili dok određeni uvjet nije ispunjen. Postoji i jako puno situacija iz svakodnevnog života koje podsjećaju na petlje: prikazivanje množenja kao uzastopnog zbrajanja, u brojalicama kada rukom pokazujemo na suigrače dok god traje izgovaranje teksta. Učenici se sigurno mogu sjetiti još primjera iz svakodnevnog života. Prisjetimo se kako u svakodnevnim životnim situacijama imamo jako puno prilika kada istu radnju moramo izvršiti više puta (npr. složiti knjige u policu, sakupiti rasute autiće u kutiju za igračke...). U takvim situacijama gotovo uvijek tražimo način kako da te, stalno iste, dosadne radnje izvršimo na što lakši, brži i elegantniji način. Zato pokušamo što više knjiga/autića odjednom vratiti na policu/ u kutiju.

Tako smo i u Matematici za uzastopno zbrajanje istog broja „izmislili“ množenje, pa tako ne moramo više računati $2 + 2 + 2 + 2 + 2$ već jednostavno 5×2 .

I kod crtanja likova imamo isti problem! Učiteljica predlaže učenicima da svom ljudskom robotu napišu program koji će crtati kvadrat. Najprije bez, a zatim s petljom. Koji bismo još geometrijski lik mogli nacrtati uporabom petlje? Koliko puta treba ponavljati naredbe za crtanje?

Učenici otvaraju stranice Code.org i prolaze redom lekcije na Nivou 6 i 7

<https://studio.code.org/s/course2>

Uspješno riješen zadatak spremaju u svoj *e-portfolio* kao snimku zaslona (ne moraju spremiti sve zadatke, ali barem 2-3 i također poveznicu na sam tečaj). Na kraju ove aktivnost s petljama učenici bi trebali u svom *e-portfolio* imati zabilježen tijek upoznavanja s petljom.

Za slijedeću aktivnost od učenika se očekuje da primijene naučeno iz cjeline Programiranje i izrade program koji koristi petlje i odluke. Trebaju osmisliti i opisati postupak rješavanja jednostavnih problema, analizirati postupak rješavanja i ispravljati pogreške. Učenici bi trebali razvijati suradničke vještine i logičko zaključivanje, te primijeniti naučene strategije rješavanja na novim zadacima.

Aktivnost 5 – Uvjeti i odluke u programiranju

Učiteljica najavljuje igru programiranja u kojoj ćemo se zabaviti i primijeniti naučeno iz cjeline Programiranje. U svemu tome će nam pomoći pčelica koja želi obići cvjetove i sakupiti nektar kako bi napravila med. Svaki cvijet koji pčelica treba obići nije isti – neki sadržavaju manje, a neki više nektara, pa će se pčelica do takvih cvjetova morati vratiti više puta. Pčelica želi da mi koristimo što manje blokova – zato ćemo, umjesto višekratnog ponavljanja istih naredbi, koristiti ružičaste blokove i kreirati petlje. Ponekad cvjetovi uopće nemaju nektara. Zato će pčelica ponekad trebati ispitati ima li cvijet uopće nektara, a ako ima donijet će odluku da ga sakupi. Za te ćemo radnje koristiti tamnoplavi blok za ispitivanje uvjeta i donošenje odluka. Pčelica će nam uvijek pomoći savjetom pa moći ćemo ispravljati svoje pogreške onoliko puta koliko nam bude potrebno za točno rješavanje zadatka.

Učenici otvaraju stranicu <https://studio.code.org/s/course2>, **Tečaj 2**, rješavaju nivo 13.

Nakon što su došli do točnog rješenja, blokove s kodovima i svoja zapažanja bilježe u *e-portfolio*. Za svaki zadatak stavljaju novi slajd u prezentaciju, a sliku rješenja kopiraju sa zaslona (Umetanje > Slika zaslona > Zaslonski isječak u PowerPointu).

Aktivnost 6 – Veliki problem > mali problem > nema problema!

Svi smo se ponekad našli u situaciji kada smo trebali riješiti neki problem, a nije bilo nikoga u tom trenutku oko nas tko bi nam mogao pomoći. Neka se učenici prisjete takvih situacija: kod pisanja zadaće, otkrivanja kako se prelazi nivo tek instalirane igrice, kako upravljati novim, tek kupljenim pametnim telefonom ili nekom drugom pametnim uređajem. U ovom ćemo projektu vježbati sa svojim prijateljima kako sebi pomoći u takvim situacijama. Prisjetit ćemo se da smo i kod rješavanja zadataka u Digitalnoj akademiji shvatili da je za uspješno rješavanje na prvi pogled ne rješivih problema najvažnije postupiti prema sljedećem obrascu:

- rastaviti problem na više malih problema
- uočiti sličnosti: dijelove zadatka / uređaja koji su nam otprije poznati
- izdvojiti te poznate dijelove kako bismo došli do novog, nepoznatog dijela zadatka i fokusirati se na rješavanje tog dijela
- stvoriti niz uputa koje će nam pomoći da u sličnoj situaciji i mi, ali i svi drugi koji bi slijedili te upute, mogu riješiti sličan problem

Učenici će se prijaviti na platformu Code.org (<https://code.org/>), odabrati Tečaj 3 > Nivo 1: Razmišljati poput računala: igre 1 i 2.

U 1. igri učenici će uočiti međusobne sličnosti tri rečenice iz zadatka: sve imaju po 4 riječi, druga riječ je upravo riječ **je**, te uočiti koje od nepotpunih rečenica ne odgovaraju tom uzorku: prva ima 5 riječi, treća ima 3 riječi, četvrta ima određenu prvu riječ koja nije zajednička svim primjerima u zadatku. Slijedeći ovakav tijek razmišljanja nameće zaključak da je jedini prihvatljiv uzorak ima druga nepotpuna rečenica prikazano na slici


Učiteljica podijeli učenike u parove i potiče ih da argumentiraju i artikuliraju tijekom razmišljanja kod rješavanja 2. igre.

Slika 12. Zadatak spajanje parova i rješenje

Spoji parove

Sva ova vozila nastavljena su na navedeno dijelove. Spoji vozila s njihovim postatima dijelova.

Počaj!



Kolači/strana: 2 Prozori/strana: 0	?	
Kolači/strana: 2 Prozori/strana: 1	?	
Kolači/strana: 2 Prozori/strana: 6	?	
Kolači/strana: 3 Prozori/strana: 2	?	

Počaj!

Spoji parove

Sva ova vozila nastavljena su na navedeno dijelove. Spoji vozila s njihovim postatima dijelova.

Počaj!

Kolači/strana: 2 Prozori/strana: 0	
Kolači/strana: 2 Prozori/strana: 1	
Kolači/strana: 2 Prozori/strana: 6	
Kolači/strana: 3 Prozori/strana: 2	

Počaj!

Izvor.: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

Učiteljica ponudi zatim parovima učenika slijedeći logički zadatak:

$$\mathbf{JA + JA + JA + JA = OJ}$$

Neka učenici pokušaju odgonetnuti koje znamenke trebaju zamijeniti slova u zadatku kako bi jednakost bila točna.

Tijek rješavanja na koji ih učiteljica navodi je naveden u slijedećim redovima.

Učenici trebaju uočiti sličnosti:

- svih 5 brojeva su dvoznamenkasti brojevi
- sva četiri pribrojnika su isti brojevi
- znamenka desetica pribrojnika i znamenka jedinica zbroja su zapravo ista znamenka

Uzimajući u obzir ove sličnosti i izlučujući uočen problem: znamenku A i O, dolazimo do zaključka da:

- J jedino može zamijeniti znamenka 1 ili 2 inače bi slovo O trebalo predstavljati dvoznamenkasti broj ($4 \times J = O$)
- ukoliko je $J = 1$, onda A ne možemo odrediti jer ne postoji broj koji zbrojen 4 puta u jedinicama zbroja ima 1
- ukoliko je $J = 2$, onda $A = 3$ i $O = 9$; što znači da je točno rješenje: **$23 + 23 + 23 + 23 = 92$**

Aktivnost 7 – Zombi i suncokret u labirintu

Učiteljica najavljuje nastavak programerskih aktivnosti i u ovom projektu. Učenici će se prijaviti na platformu Code.org, odabrati Tečaj 3 i Nivo 2: Labirint. U 15 ponuđenih igara

učenici će pomagati malenom zombiju da dođe do suncokreta. U prvih nekoliko igara ponavljamo slaganje naredbi za kretanje naprijed i okretanje lijevo – desno, dok se od 4. igre pojavljuje potreba korištenja petlje (kako bismo smanjili potrebu nizati iste naredbe više puta). Od 9. igre pojavljuje se novi blok – petlja: „repeat until“ ili „ponavljaj dok se ne ispuni određeni uvjet“. Učiteljica može pokazati učenicima 9. igru kako bi učenici uočili kako umetanje naredbe „idi naprijed“ u ovaj blok dovodi do toga da će zombi ići naprijed dok god ne stigne do suncokreta, a da uopće ne moramo brojati koliko je koraka zombiju potrebno da dosegne suncokret:

Slika 13. Korištenje petlje „repeat until“



Izvor: Projekt 1 – Početak nove priče - rad učenika 3. razreda (Algebra, Digitalna akademija)

Slično će postupiti i u ostalim igrama. U 14. igri će učenici pokušati upariti svaki zadatak (lijevo) sa složenim blokovima naredbi (desno).

Slika 14. Zadatak i rješenje spajanja parova Labirinta



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

U 15. igri učenici trebaju između 4 ponuđena odabrati one blokove naredbi koji rješavaju predstavljeni zadatak. Učenici će se prisjetiti kako zadatak treba rastaviti na manje dijelove, uočiti sličnosti (sva rješenja imaju iste, točno složene petlje na početku

i na kraju), izdvojiti razlike ko što su da nakon prvotnog kretanja naprijed zombi mora skrenuti desno; 2 koraka a ne 1 prethodi novom skretanju desno.

Slika 15. Višestruki točan izbor - kviz pitanje



Izvor: Projekt 1 – Početak nove priče –učenici 3. razreda (Algebra, Digitalna akademija)

Aktivnost 8 – Pčela i funkcije

Učiteljica ponovno poziva učenike na platformu Code.org gdje ćemo tijekom rada na 11 igara Nivoa 6 upoznati **funkcije**. Funkcije predstavljaju izdvojene programske cjeline, svojevrsne potprograme, koji ulazne podatke pretvaraju u nove podatke (izlazne podatke). U funkcijama definiramo svoj niz naredbi, te ga imenujemo. Funkcije su u Code.org prikazane kao blokovi jasno zelene boje na kojima je istaknut naziv funkcije što je prikazano na slijedećoj slici.

Slika 16. Zadatak korištenja funkcija



Izvor: Projekt 1 – Početak nove priče – učenici 3. razreda (Algebra, Digitalna akademija)

Ako se želimo podsjetiti koje točno naredbe sadržava blok funkcije, možemo pogledati u sivi okvir naslova Funkcija kao što je pokazano na slijedećoj slici.

Slika 17. Prikaz nove get funkcije



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

U Nivou 6 pojavljuju se blokovi funkcije „get“ i „move and get“ koje treba kombinirati kako bi se riješio zadatak. U 7. igri treba stvoriti vlastitu funkciju koja je već imenovana, samo za nju treba složiti odgovarajuće naredbe. U 9. igri stvoriti funkciju kako bi pčela sakupljala med jedino iz cvjetova koji imaju nektar (oznaka 0 je za one bez, a oznaka 1 za one cvjetove koji imaju nektar).

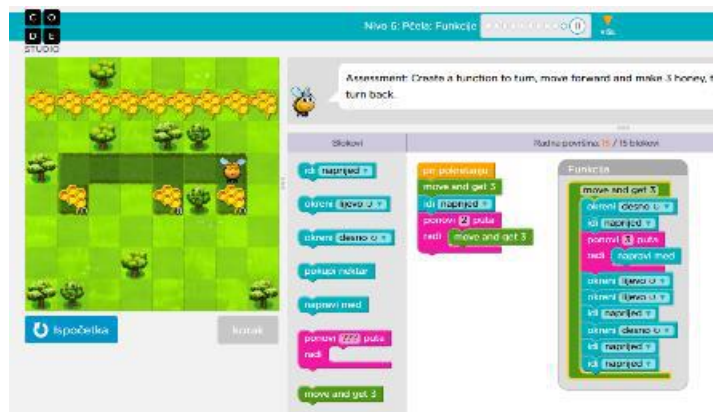
Slika 18. Prikaz "collect purple nectar" funkcije



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

U 11. igri trebamo stvoriti funkciju koja će omogućiti pčeli da se nakon izrade 3 meda svaki puta vrati na mjesto odakle kreće. Slijedi prikaz rješenja 11. igre.

Slika 19. Prikaz nove funkcije



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

Nakon što svi učenici završe sa izradom svih 11 igara Nivoa 6, učiteljica potiče učenike da komentiraju svoje uratke: navedu koji im je nivo bio najzahtjevniji i zašto, te kako tekao postupak rješavanja. Koju su pogrešku napravili? Što su iz nje naučili? Neka drugi učenici ponovno otvore istu igru koja se upravo komentira i pogledaju jesu li oni možda tu igru riješili na drugačiji način. Tko je napisao najmanje linija koda? Koliko nam korištenje blokova petlji i funkcija smanjuje broj linija koda u glavnom „main“

programu? Koliko bi linija koda sadržavao pojedini program (rješenje) da ne koristimo blok petlje i funkcije?

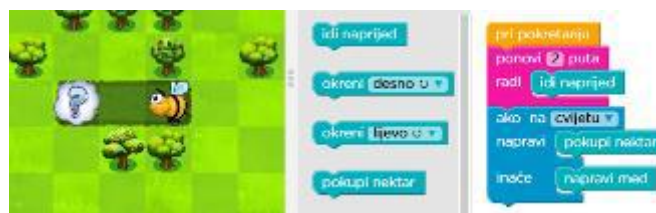
U slijedećoj aktivnosti učenici će Izrađivati programe koristeći se nizovima naredbi, odlukama i petljama. Naučiti će koristiti različite programe kojim će se izrađivati novi sadržaj, te rješavati problem. Koristiti se računalom za unaprjeđivanje tehnike učenja i zapamćivanja, te razvijati osobne potencijale.

Aktivnost 9 – Kako otkriti skriveno?

Učiteljica najavljuje nastavak Code.org pustolovina sa pčelicom koja želi sakupljati nektar i od njega stvarati med. Danas ćemo pomagati pčelici u Nivou 7 Tečaja 3. Pčelici će ovoga puta zaista trebati sva pomoć jer danas obilazi cvjetove za koje se uopće ne zna imaju li nektara ili nemaju. Kako bismo osigurali ispunjavanje zadatka bez obzira što ne znamo potreban broj ponavljanja radnje prikupljanja nektara, uvest ćemo novi blok s uvjetima „if“: oni nam omogućavaju da se radnja odvija sve dok se ne ispuni uvjet tj. prikupi sav nektar ako ga uopće ima. Najbolje da učenici isprobaju funkcioniranje novog bloka odigravajući prve 3 igre Nivoa 7.

Neka učenici objasne kako su u ove 3 igre koristili „if“ blok i koliko nam je on pomogao – kako bismo inače ispitati ima li cvijet nektara ili ne? U 4. igri imamo novi blok „if - else“ koji nam omogućava ispitati uvjet i izvršiti ga, a u suprotnom izvršit ćemo drugu naredbu. Slijedi prikaz jednog od rješenja 4. igre.

Slika 20. Prikaz rješenja zadatka koristeći if-else pelju



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

7. – 9. igra su nešto su složenije i zato učenicima za njih treba dati malo više vremena. U 7. igri uvodimo funkciju kojom ćemo ispitati koliko koji cvijet ima nektara kako bismo sve pokupili. Slijedi prikaz jednog od mogućih rješenja.

Slika 21. Funkcija kojom ispitujemo koliko koji cvijet ima nektara



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

U 8. igri trebamo složiti funkciju kojom ćemo ispitati koliko nektara cvijet ima, sakupiti sav nektar a onda i napraviti med. Slijedi prikaz jednog od mogućih točnih rješenja.

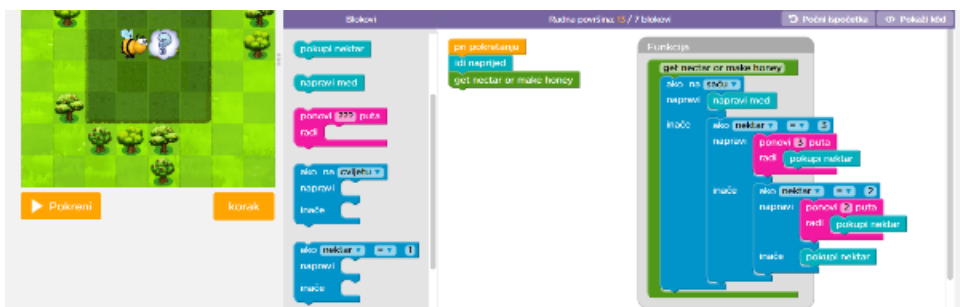
Slika 22. Prikaz rješenja 8. zadatka



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

U 9. igri trebamo ispitati skriva li oblak cvijet sa 1 nektarom ili skriva saće koja treba 1 med. Trebamo složiti funkciju „pokupi nektar ili napravi med“, a zatim pokupiti nektar odnosno med. Na slijedećoj slici prikazan je jedan od mogućih točnih rješenja.

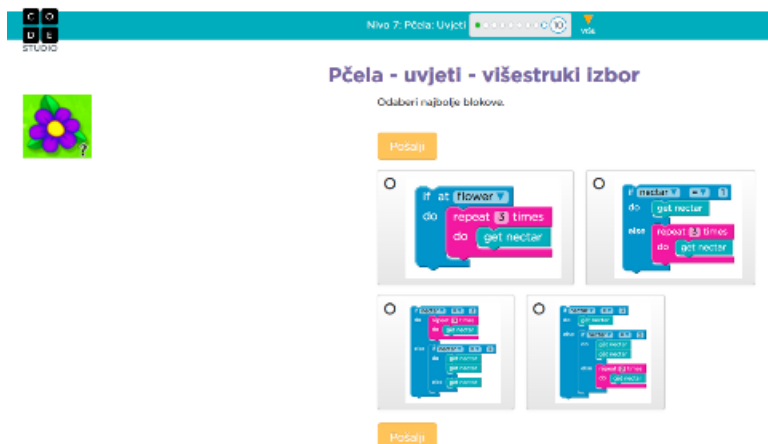
Slika 23. Funkcija „pokupi nektar ili napravi med“



Izvor: Projekt 1 – Početak nove priče - rad učenika 3. razreda (Algebra, Digitalna akademija)

U 10. će igri učenici odabrati najbolje složen „if“ blok kojim će se ispitati koliko cvijet na slici ima nektara. Točan je 3. odgovor prikazan na slici u nastavku.

Slika 24. Kviz ispitivanje uvjeta



Izvor: Projekt 1 – Početak nove priče - učenici 3. razreda (Algebra, Digitalna akademija)

Aktivnost 10 – Ugniježdene petlje

Učiteljica najavljuje nastavak pustolovina s našom pčelicom. Osim što ćemo pomoći pčelici da uspješno obavlja svoj posao, ponoviti ćemo sve vrste naredbi koje smo do sada učili i koristili, kao i sve vrste petlji. Učenici se prijavljuju na Code.org Tečaj 3, Nivo 13: Ugniježdene petlje gdje ćemo odigrati svih 13 igara. U prvoj igri koristimo petlju kako bismo ispisali što manje linija koda, a pčelica sakupila nektar iz svih cvjetova. Od 2. do 5. igre koristimo ugniježdene petlje: kada petlju sa različitim naredbama umetnemo u drugu petlju kojom odredimo koliko će se puta naredbe iz ugniježdene petlje ponavljati:

Slika 25. Ugniježdene petlje



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

U 6. igri trebamo „while“ petlju kojom osiguravamo ispunjenje naredbe bez obzira na činjenicu da ne znamo koliko točno puta trebamo izvršiti određenu naredbu. Petlja „while“ izvršavat će naredbu dok se uvjet ne ispuni.

Slika 26. Prikaz rješenja korištenjem while petlje i uvjeta



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

Slijedi prikaz najsloženijeg zadatka ovog nivoa u kojem su korištene ugniježdene petlje, “while” petlje, “repeat” petlje i uvjeti.

Slika 27. prikaz rješenja korištenjem svih naučenih petlji i uvjeta



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

Aktivnost 11 – Ispravimo pogreške

Učiteljica potiče učenike da ispričaju kada su posljednji put napravili neku pogrešku, kako su je ispravili i što su iz nje naučili. Zaključujemo kako je praviti pogreške normalno i dobro. One su jasan pokazatelj da si pokušao napraviti nešto novo. Neka se učenici osvrnu oko sebe: sve što vide nastalo je kao posljedica nečije želje da se napravi nešto novo, bolje, udobnije, brže, korisnije. Sve stvari koje nas okružuju nastale su kao posljedica ispravljanja brojnih pogrešaka. Put od ideje do konačnog rezultata je jako dug, pun poteškoća i frustracija zbog silnih neuspjeha, ali nas na kraju tog puta, ako smo dovoljno uporni, čeka nagrada: uspjeh i ostvarenje našeg cilja! Kažu da je Edison napravio prvu uspješnu žarulju nakon 10 000 neuspješnih pokušaja! Koliko smo mu zahvalni što je bio toliko uporan!

Zato ćemo mi danas njegovati ovu vrijednu osobinu: upornost - tražeći pogreške i ispravljajući ih. Učenici će se prijaviti na Code.org > Tečaj 3 > Nivo 14 i u nizu od 12 igara ćemo pokušavati uočiti pogrešku, a zatim je i ispraviti. Učenici mogu uočiti pogrešku tako da pokrenu već napisani program, uoče mjesto gdje se pogreška potkrala te zamijene/dodaju blokove koji ometaju pravilno izvršavanje programa.

U 13. igri učenici trebaju odrediti koja inačica funkcije će raditi ispravno, a točno je 2 drugo od ponuđenih rješenja.

Slika 28. Kviz pitanje ispravljanje greški u funkcijama



Izvor: Projekt 1 – Početak nove priče – rad učenika 3. razreda (Algebra, Digitalna akademija)

Aktivnost 6 – Primijenimo naučeno u programiranju

Učiteljica najavljuje niz aktivnosti u kojima ćemo slažući programske blokove ponoviti najvažnije situacije iz programiranja koje smo do sada naučili rješavati: kako blokom za petlje smanjiti broj korištenih blokova tj. napisanih linija koda; kako ispitati različite uvjete, kada i kako koristiti ugniježdene petlje i funkcije. Učenici mogu pokušati definirati nabrojane pojmove odnosno pokušati navesti primjere situacija kada je neke od njih korisno i potrebno koristiti. Učiteljica ponavlja sa učenicima što su to petlje, ugniježdene petlje, funkcije i uvjeti.

Petlje: naredbe koje se umetnu u ove blokove ponavljat će se onoliko puta koliko smo ponavljanja na bloku za petlje i naveli.

Primjer za petlju: umjesto nizanja 20 blokova za kretanje jedan po jedan korak naprijed uzmemo samo 1 blok za petlju na kojem navedemo broj 20 koji označava broj ponavljanja, pa u takav blok za petlje umetnemo samo jedan blok za kretanje naprijed.

Ugniježdene petlje: kada petlju sa naredbama umetnemo u drugu petlju kojom odredimo koliko će se puta naredbe iz ugniježdene petlje ponavljati.

Primjer za ugniježdene petlje: ako trebamo nacrtati 3 reda cvjetova od kojih svaki ima po 5 jednakih latica: najprije u prvi blok za petlju naznačimo broj 5 i umetnemo jedan blok za crtanje latica, a zatim taj blok umetnemo u još jedan blok za petlje na kojem naznačimo 3 ponavljanja crtanja cvijeta (umetnemo i blok za kretanje kako se crtanje ne bi ponavljalo na istom mjestu)

Funkcije: izdvojene programske cjeline, svojevrsni potprogrami, koji ulazne podatke pretvaraju u nove podatke (izlazne podatke). U funkcijama definiramo svoj niz naredbi, te ga imenujemo. Jedan funkcijski blok unutar programa zamjenjuje taj niz naredbi, a prepoznamo ga po imenu koje nosi.

Primjer za funkcije: Ukoliko crtanje latice cvijeta iz prethodnog primjera traži više naredbi – stvorimo funkcijski blok koji nazovemo „latica“, pa njega umetnemo u ugniježđenu petlju.

Ispitivanje uvjeta: Ovi su blokovi imenovani sa *if*(ako je) ili *if-else* (ako je/inače). Naredbe unutar njih ponavljat će se onoliko puta koliko je potrebno dok se uvjet ne ispunji. U „*if-else*“ bloku određujemo i što svaki put učiniti kada uvjet nije ispunjen.

Primjer za IF: ako je crveno na semaforu za pješake – stani

Primjer za IF/ELSE: ako je crveno na semaforu za pješake – stani / inače - prijeđi ulicu

Učenici će se prijaviti na platformu Code.org (<https://code.org/>), odabrati Tečaj 3 > Nivo 3: Umjetnik i to od 1. do 12. igre.

U svim je igrama potrebno nacrtati simetrične likove kod kojih imamo ponavljanje uzorka: pravokutnike, jednako-stranične trokute usporedne crte i sl. Učenici imaju bogat izbor različitih blokova koji, osim samog kretanja, omogućuju programirati različite boje, debljine crta, preskoke i sl. Jedini je uvjet koristiti što manje blokova, što znači da je uputno koristiti ružičaste blokove za oblikovanje petlji (ali i ugniježđenih petlji) i tako izbjeći nepotrebno ponavljanje istih radnji što je vidljivo na slici u nastavku.

Slika 29. Zadatak crtanja simetričnog cvijeta uz slaganje blokova kodova



Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

Na kraju učiteljica zajedno sa učenicima analizira način rješavanja zadatka tako što će se dogovoriti da svi zajedno otvore jednu od igara i usporedimo način rješavanja

kod različitih učenika. Na pokazanom primjeru rješavanja 10. igre vidljivo je kako je dvoje učenika odabralo različite strategije: jedan se odlučio najprije nacrtati ravnu crtu po cijeloj visini uzorka, dok se drugi odlučio za pojedinačno crtanje trokuta. Iako su oba rješenja točna jer vode do uspješnog ispunjavanja zadatka, proglašavamo drugo rješenje kao uspješnije jer je kod rješavanja iskorišteno manje blokova odnosno linija koda.

Slika 30. Primjer optimalnog rješenja 10. zadatka



Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

Za kraj ove aktivnosti frontalno odigramo 13. i 14. igru gdje imamo izbor pravog programa za crtanje predloženog lika između 4 ponuđena programa. Neka učenici objasne svoj izbor točnog odgovora, ali neka i objasne svoje razmišljanje tj. postupak eliminacije ostalih odgovora.

Za slijedeću aktivnost učenici trebaju osmisliti i izraditi novi sadržaj kojim se povezuju sadržaji iz različitih područja, rješavati problem i predstavljati njegovo rješenje. Učenici će razvijati interes za STEM područje: mehaniku, robotiku i programiranje, obnovljive izvore energije. Analizirati će jednostavan logički zadatak, osmisliti i grafički prikazati rješenje jednostavnog problema.

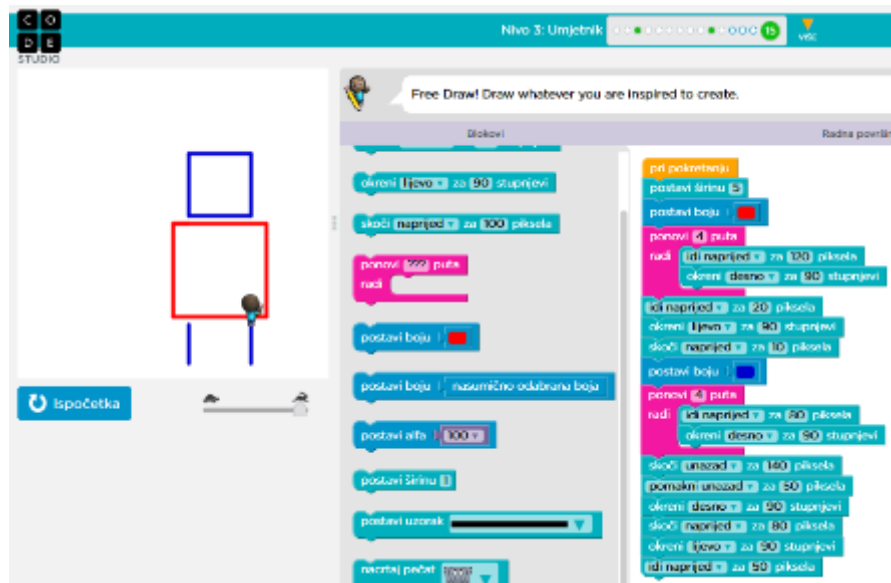
Aktivnost 12 – Roboti u 3. razredu

Učiteljica potiče učenike da se prisjete (pregledavajući *e-portfolio*) u kojim smo se sve projektima/aktivnostima susretali sa robotima: učenici će se prisjetiti da smo se u 5. projektu imali prilike susresti sa nekoliko robota. Neka u svom prezentacijskom *portfoliju* otvore novi slajd, imenuju ga sa Roboti u 3. razredu, pa pronađu slike svih robota koje smo do sada susretali, kopiraju ih i zalijepe na ovaj novi slajd. Ne bismo smjeli zaboraviti uz svaku sliku dodati izvor: od kuda smo slike preuzeli jer smo tijekom ovih projekata naučili kako se treba odnositi prema autorskim pravima sadržaja koje pronalazimo na internetu. Učiteljica naglasi i kako je potrebno na ovom slajdu ostaviti dovoljno mjesta za još jednog robota kojeg će upravo sada stvoriti.

Učiteljica sa učenicima najprije ponovi što smo na prošlom susretu radili u Code.org te najavljuje zabavnu aktivnost u preostaloj, 15. igri Tečaja 3, Nivoa 3: Umjetnik. U toj igri omogućen nam je slobodan izbor teme/cртеža koji možemo programirati, a učiteljica će objasniti kako će učenici pokušati složiti blokove kako bi se nacrtao robot, i to po uzoru na robota kojeg smo u 5. projektu slagali od različitih dijelova. Sada, u 7. projektu 3. razreda Digitalne akademije učenici znaju sami crtati robota koristeći blokovsko programiranje! Neka učenici još jednom pogledaju robote koje su zalijepili na ovom posljednjem prezentacijskom slajdu kako bi analizirali od kojih se dijelova tijelo robota sastoji (uglavnom kvadrati i pravokutnici), te koji bi im blokovi i naredbe mogli zatrebati u programiranju (petlje, preskoci...).

Sada će se učenici prijaviti na platformu Code.org (<https://code.org/>), odabrati Tečaj 3 > Nivo 3: Umjetnik > 15. igra; pa neka pristupe kreativnom stvaranju programa. Mogu koristiti i blok za promjenu boje i/ili uzorka crte. Slijedi slika jednog primjera nacrtanog robota koristeći dostupne kodne blokove za crtanje.

Slika 31. Rješenja zadatka crtanja robota slaganjem kodnih blokova



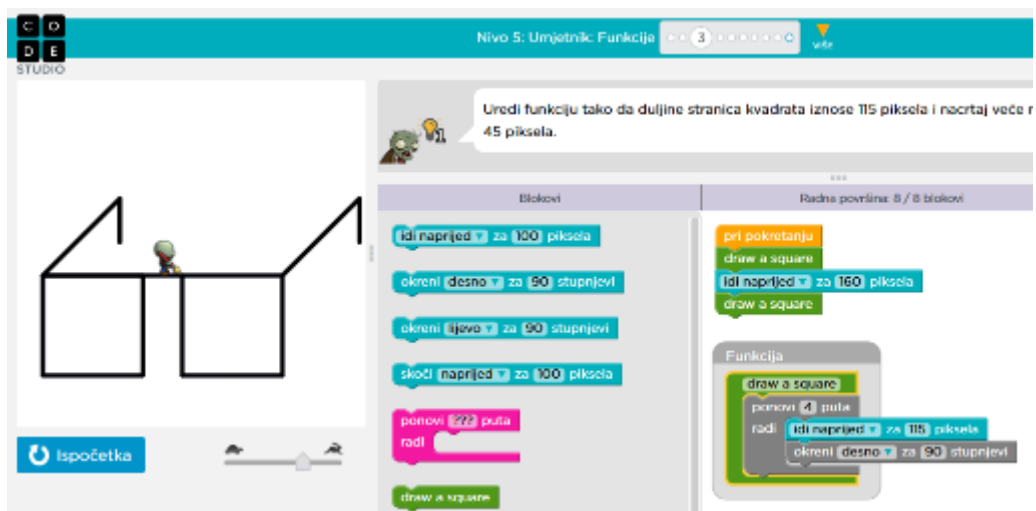
Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

Učenici trebaju napraviti snimak zaslona svog programiranog uratka, pa ga zalijepiti u slajd e-mape pored njegovih ostalih prijatelja robota. Mogu dodati i snimak svog programa na koji su zacijelo vrlo ponosni! Neka učenici međusobno pregledaju prezentacijske slajdove jedni drugima, prokomentiraju ih i pohvale.

Aktivnost 13 – Funkcijski blok

Učiteljica sa učenicima treba ponoviti što kod programiranja predstavlja (zeleni) funkcijski blok. Najčešće se koristio kod crtanja različitih oblika koji se sastoje od više pravilnih geometrijskih likova, pa će i nadalje za crtanje različitih geometrijskih, gradbenih elemenata crteža koristiti taj funkcijski blok. Do sada se pokazalo kako je kod slaganja naredbi nužno precizno i točno definirati funkcijski blok i za tu je aktivnost uputno potrošiti dovoljno vremena i kreativnosti. Učenici će na platformi Code.org odabrati Tečaj 5 > Nivo 5: Umjetnik: Funkcije i to od 1. do 7. igre, te primijeniti naučeno o upotrebi funkcijskih blokova.

Slika 32. Funkcija "draw a square"



Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

Učenici trebaju riješiti prvih 7 igara. Trebaju napraviti snimak zaslona 6. i/ili 7. igre i spremiti ih na novi slajd svoje prezentacije. 8. igra: Crtanje pahuljice dodavanjem i mijenjanjem već postavljenih naredbi unutar funkcije „draw a snowflake“ neka posluži učenicima koji su među prvima gotovi da dodatno razviju svoju kreativnost i snalaženje kod slaganja programskih blokova, a da usput daju priliku i manje spretnim učenicima da dovrše prvih 7 zadataka. Kada su svi učenici gotovi, zajednički proanalizirajmo i usporedimo rješenja 6. i 7. igre: utvrdimo koju su strategiju rješavanja učenici koristili i tko je riješio zadatke koristeći najmanje programskih blokova.

Aktivnost 14 – Zupčanici u 3. razredu

Učiteljica ponavlja što smo sa učenicima tijekom ovog razreda naučili o obnovljivim izvorima energije: energiji vjetra i sunca. Učiteljica učenicima pokazuje početnu stranicu igre sa <https://www.learn4good.com/games/kids/renewableenergy.htm> kako bi se učenici prisjetili kako smo u igri pomogli graditi podzemni stroj koji je ekološki osviještenom naselju pomogao stvarati električnu energiju kada nije bilo vjetra pa vjetrenjače nisu mogle proizvoditi.

Neka netko od učenika pokuša objasniti gdje se još mogu pronaći ovakvi mehanizmi (npr. na biciklu, starim satovima, itd.), pa neka pokušaju objasniti osnovne principe rada ovakvih mehanizama: koji se zupčanici brže vrte (mali), ako se jedan vrti udesno – na koju se stranu vrti susjedni (ulijevo) i sl.

Učenici u novom prozoru otvaraju 9. igru 5. nivoa Code.org na kojem smo već radili kod prethodne aktivnosti: u toj su 9. igri već predložene dvije funkcije za crtanje velikog i malog zupčanika.

Slika 33. Funkcije za stvaranje zupčanika



Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

Neka se učenici potrudite uklopiti ove funkcijske blokove u stvaranje većeg stroja – sa više zupčanika. Neće crtati bilo kakav mehanizam nego trebaju nacrtati mehanizam zupčanika koji će nalikovati jednom od prvih nivoa igre sa podzemnim zupčanicima kojoj će pristupiti slijedeći poveznicu koju im je učiteljica poslala novom porukom elektronske pošte naslovljena sa Pokreni vjetrenjaču. Cilj je ove aktivnosti izraditi novi slajd u svojoj e-mapi naslova Zupčanici u 3. Razredu na koji će učenici zalijepiti snimak zaslona odigrane igre i sličan stroj kreiran pomoću blokova kodova (primjer takvog slajda je na slici u nastavku).

Slika 34. Prezentacijski slajd radova učenika na temu Zupčanik



Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

Neka učenici sami odrede što bi bilo pametnije prvo napraviti: odigrati jedan nivo igre ili pristupiti programiranju (kod pokretanja već složenog programa učenici će uočiti kako je u Code.org programu mali zupčanik unutar velikog, pa će ih najprije trebati razdvojiti).

Učenici će, po završetku, usporediti uratke, objasniti što im je kod programiranja bilo najzahtjevnije i koliko su im u radu pomogli funkcijski blokovi – koliko bi tek program bio dug i nepregledan da je trebalo za svaki zupčanik ispočetka slagati sve naredbe! Ovakav prekrasan slajd također zaslužuje biti pokazan roditeljima na završnoj svečanosti!

Za kraj neka učenici odigraju i posljednju, 10. Kviz igru ovog nivoa. U njoj imamo mogućnost višestrukog izbora za predloženu funkciju kojom bi se nacrtali prikazani dvostruki pravokutnici.

Učenici do točnog rješenja mogu doći i nasumičnim odabirom, stoga je iznimno važno proanalizirati ponuđena rješenja: neka učenici objasne zbog čega je prvi ponuđeni odgovor točan i zbog čega bismo odnosno koji bismo odgovor najprije odbacili kao netočan (drugi jer uopće ne predviđa crtanje drugog pravokutnika, a posljednji jer uopće nije predviđen nikakav razmak između crtanja dvaju pravokutnika).

Za slijedeću aktivnost učenici će naučiti analizirati postavljene zadatke, prepoznati situacije u kojima se koriste petlje i odluke, pa uporabom različitih naredbi izraditi svoj uradak koristeći se jednostavnim okruženjima za programiranje.

Aktivnost 15 – Ispitajmo uvjete

Učiteljica najavljuje ponovno druženje sa zombijima i suncokretima. Kao što smo to već otprije saznali zombi obožava suncokrete, a mi mu slaganjem blokova trebamo pomoći da se u labirintu domogne svoje poslastice. Ponovit ćemo kako kod kretanja kroz labirint nije važno odrediti točan broj koraka za napredovanje, koliko je važno odabrati put kojim se izbjegavaju zamke i prepreke te ga definirati koristeći IF odnosno IF/ELSE blokove.

Sada će se učenici prijaviti na platformu Code.org, odabrati Tečaj 3 > Nivo 8: Labirint: Uvjeti, pa početi rješavati prve od 12 igara ovog nivoa. Cilj svih igara je jednak: trebamo slagati programske blokove kako bismo pomogli zombiju u labirintu da stigne do suncokreta. Kod programiranja se trebamo potruditi koristiti što manji broj blokova, a u tome će nam pomoći „if“ (ako je) i „if-else“ (ako je/inače) blokovi kojima ispitujemo mogućnosti kretanja desno, lijevo ili naprijed – ovisno o potrebama zadatka. Stoga je najvažnije da prije rješavanja svake igre učenici dobro promotre labirint i utvrde koje nam je skretanje zapravo potrebno za uspješno rješavanje zadatka odnosno koje ćemo kretanje zapravo ispitivati i tražiti u labirintu kao poželjno kretanje. Učiteljica najprije sa učenicima zajednički analizira zadatak, pa ga učenici pokušaju riješiti, a zatim učenici predstavljaju točna rješenja. Svaki se zadatak, može riješiti na više načina, no učiteljica pojasni kako su optimalna rješenja ona kod kojih se koristio najmanji mogući broj blokova.

U osmoj igri prednost dajemo kretanju naprijed jer tako izbjegavamo nevolje, poželjno skretanje je ulijevo, ali ne uvijek: skretat ćemo ulijevo samo kada nemamo mogućnost kretanja naprijed kako je prikazano rješenju ispod.

Slika 35. Rješavanje zadatka koristeći „repeat-until“ petlju



Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

U devetoj igri učenici sami slažu sve blokove. Opet dajemo prednost kretanju naprijed, ali sada skrećemo udesno samo onda kada nemamo mogućnost kretanja naprijed.

U desetoj za kretanje kroz labirint je primarno kretanje naprijed, ispitujemo mogućnost skretanja udesno – u protivnom skrećemo ulijevo.

Slika 36. Rješenje zadatka koristeći petlje i uvjete



Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

U jedanaestoj ponovno učenici samostalno slažu sve blokove. Kod analize labirinta utvrdili smo da zapravo uopće ne trebamo i ne želimo ispitivati skretanje ulijevo jer se ulijevo kriju same nevolje. Zato opet dajemo prednost kretanju naprijed, a skrećemo udesno samo onda kada nemamo mogućnost kretanja naprijed. U 12. igri izabiremo između 4 ponuđene skupine blokova: koji od njih rješavaju problem kretanja kroz labirint prikazan gore lijevo.

Slika 37. Zadatak kviz pitanje za ispitivanje uvjeta u labirintu



Izvor: Projekt 7 – Na kraju razreda – rad učenika 3. razreda (Algebra, Digitalna akademija)

Učiteljica će inzistirati na eliminaciji ostalih odgovora tj. traženju pogreške u slaganju programskih blokova o ostalim ponuđenim rješenjima: u 2. i 4. ponuđenom odgovoru već na prvom skretanju je nemoguće skrenuti desno ako je jedina mogućnost skretanja ulijevo. U 3. odgovoru uopće nemamo blok za kretanje naprijed, pa će se zombi samo okretati u mjestu.

ISTRAŽIVANJE

Kako bi se dopunilo tradicionalno poučavanje, u posljednjih nekoliko godina u Hrvatskoj je došlo do neprekidnog rasta primjene ICT (engl. *Information communication technology*) aplikacija u školstvu i razvoja IT(engl. *Information technology*) tečajeva izvan javnog školstva. Samo nekoliko studija procijenilo je prihvaćanje takvih novih tehnologija među učiteljima. Istraživanjem se pokušalo dokučiti stav učenika informatike ka korištenju web aplikacije Code.org kao instruktivnom alatu za učenje osnova računalstva u privatnoj Ustanovi za obrazovanje Algebra u osnovnim školama diljem Hrvatske (Zagreb, Osijek, Pula). Za istraživanje se koristio model TAM (engl. *Technology Acceptance Model*). Rezultati istraživanja sugeriraju da TAM kao robusni model može biti učinkovito korišten u ovom kontekstu i pronađena je dobra prilagodba predloženom modelu podataka. Uzorak se sastojao od 114 učenika osnovnih škola koji se slažu oko korisnosti web aplikacije u nastavi informatike. Čimbenik koji ukazuje najveći potencijal u ovom istraživanju je zadovoljstvo uporabe aplikacije, na što ima značajan utjecaj učenička percepcija korisnosti i stavova prema korištenju web aplikacije u nastavi.

TAM model kao okvir istraživanja

U proučavanju prihvaćanja korisnika i upotrebe tehnologije, TAM je jedan od najcitiranijih i najčešće korištenih modela. Prema TAM-u, percipirana korisnost PU (engl. *Perceived usefulness*) i percipirana jednostavnost upotrebe PEOU (engl. *Perceived ease-of-use*) primarni su motivacijski čimbenici za prihvaćanje i korištenje novih tehnologija. PU je stupanj do kojeg osoba vjeruje da će uporaba tehnologije proizvesti bolje ishode (Davis, 1989). "Korisnost" se odnosi na sposobnost korištenja prednosti. Nasuprot tome, PEOU je percepcija o stupnju napora potrebnih za korištenje određenog sustava. U ovom slučaju, ta lakoća označava se kao slobodno razmišljanje o teškoći ili potrebnom malom trudu da bi se tehnologija koristila (Dwivedi,

Y.K.,et al, 2012.). PU varijabla temelji se na promatranju da ljudi imaju tendenciju da se koriste ili ne koriste aplikacijom u mjeri u kojoj vjeruju da će im pomoći da bolje obavljaju svoj posao (Davis 1989, str. 320). PU izravno utječe na stav prema korištenju sustava i posredno utječe na namjeru budućeg korištenja softvera BI (engl. *Behavioral intention*). Čak i ako se aplikacija percipira kao korisna, to će biti samo ako se smatra lako upotrebljivom, tj. koristi od korištenja nadmašuju napor korištenja sustava. PEOU utječe na stav prema korištenju sustava ATT (engl. *Attitude towards use*).

PEOU utječe na PU. PU također ima izravan utjecaj na BI. BI vodi do stvarne uporabe sustava. Dvije ključne varijable u TAM-u su PU i PEOU. PU je definiran iz perspektive gledište korisnika. Hoće li aplikacija poboljšati njegov ili njezin radni učinak u organizaciji u kojoj djeluje? PU utječe i na stavove prema korištenju sustava i na namjeru korištenja(BI). PEOU je varijabla koja opisuje percepciju korisnika da će sustav biti jednostavan za korištenje. PEOU utječe i na PU i na stav prema upotrebi (ATT). Davis (1989) razvija i potvrđuje mjerilo za te varijable. Jednostavni model usvajanja IT-a koji tvrdi da je korištenje IT-a temeljeno na vjerovanjima korisnika kao što su percipirana korisnost sustava (PU) i percipirana jednostavnost korištenja sustava (PEOU), te čimbenici utjecaja na njihov stav prema upotrebi (ATT) i namjere ponašanja za korištenje (BI).

Dakle, obično TAM studije mogu imati tri hipoteze povezane sa temeljnim konstruktima. Prvo, za PEOU se očekuje da će utjecati na varijable: stav prema sustavu (ATT) i percipirana korisnost PU. PEOU i PU uzeti kao nezavisne varijable mogu zajedno utjecati na stavove prema korištenju, koji su odabrani kao zavisna varijabla. Međutim, kako se istraživanje provodilo nad učenicima nižih razreda osnovne škole gdje bi njima bilo vrlo teško predvidjeti njihovu namjeru ponašanja (BI) za buduće korištenje web aplikacije isti je čimbenik zamijenjen drugim, a to je zadovoljstvo korištenjem web aplikacije SAT (engl. *Satisfaction*).

Hipoteze razvijene za čimbenike PU, PEOU i ATT su:

H1: Percipirana jednostavnost korištenja ima značajan utjecaj na stav prema korištenju računala [PEOU-> ATT]

H2: Percipirana jednostavnost upotrebe ima značajan utjecaj na Percipiranu korisnost [PEOU-> PU]

H3: Percipirana korisnost ima značajan utjecaj na stav prema korištenju web aplikacije [PU-> ATT]

Nadogradnja TAM modela

Zadovoljstvo

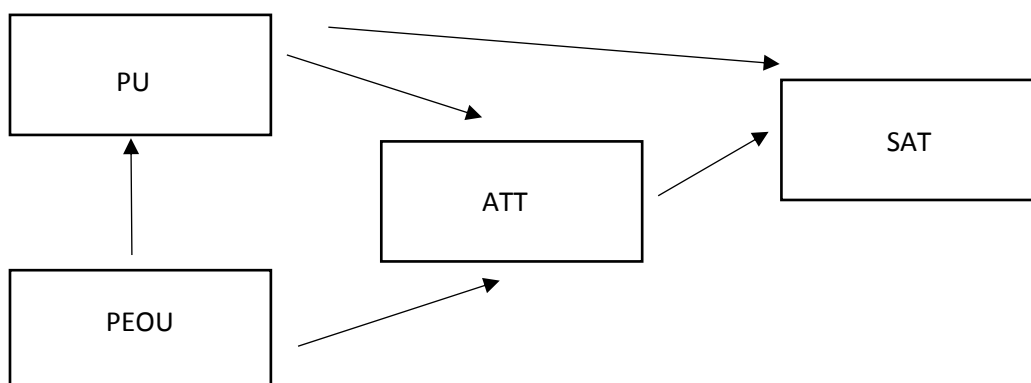
Iako je proveden velik broj studija za procjenu zadovoljstva korisnika u vezi s korištenjem sustava, bilo je vrlo teško pronaći TAM studije koje su smatrale zadovoljstvo korisnika mogućim pokazateljem namjere budućeg korištenja. Većina studija jednostavno pretpostavlja da se zadovoljstvo korisnika može očitovati prihvaćanjem i korištenjem tehnologije, zanemarujući zadovoljstvo kao varijablu ishoda. Budući da je jedan od naših ciljeva istraživanja procijeniti čimbenike koji utječu na zadovoljstvo malodobnih učenika, odlučila sam eksplicitno izmjeriti učenikovo zadovoljstvo web aplikacijom kao nadogradnju TAM modela. Prilikom učenja kodiranja, učenje se odvija putem web sučelja, pa samim time i kvaliteta komunikacije može utjecati na tehnološki sustav koji se koristi. Ako tako gledamo na sustav, očekujem da će stavovi učenika i tehnologija utjecati na zadovoljstvo. Stoga sam razvila hipoteze u nastavku.

H4: Percipirana korisnost ima značajan utjecaj na zadovoljstvo korištenjem web aplikacije [PU-> SAT]

H5: Stav prema korištenju ima značajan učinak na zadovoljstvo korištenjem web aplikacije [ATT->SAT]

Slika 39. sažima hipoteze koje su predložene u ovom odjeljku u nadograđeni TAM model autora rada.

Slika 38. Autorov prošireni TAM model



Izvor: Autorov model

Varijable modela

Razvoj PEOU, PU i ATT nezavisnih varijabli upitnika za ovo istraživanje bazira se na Davisovom modelu. Četiri nezavisne varijable korištene su za operacionalizaciju PEOU i još četiri operacije nad PU, a za ATT i SAT su postavljene po tri nezavisne varijable.

Četiri PU nezavisne varijable su:

- StudioCode mi je bio koristan prilikom učenja programiranja igrice
- StudioCode mi je olakšao da razumijem na koji način se programiraju igrice
- Kada bih nastavio/la koristiti sigurno bih naučio/la programirati igrice
- Znanje koje sam dobio/la korištenjem StudioCode sigurno će mi pomoći kada budem dalje učio/la programirati igrice

Četiri nezavisne varijable koje su operacionalizirale PEOU:

- Bilo mi je jednostavno koristiti StudioCode
- Razumio/la sam napisane upute za rješavanje zadatka na StudioCode
- Jednostavno sam koristio/la blokove za slaganje koda
- Sigurno bih jednostavno savladao/la sve slijedeće tečajeve na StudioCode

Tri ATT nezavisne varijable su:

- Mislim da je dobro koristiti StudioCode
- Mislim da je pametno koristiti StudioCode
- Sviđa mi se ideja da nastavim koristiti StudioCode

Tri nezavisne SAT varijable su:

- Dok sam koristio/la StudioCode bilo mi je zanimljivo
- Bilo mi je zabavno koristiti blokove i doći do rješenja zadatka u StudioCode
- Zadovoljan/na sam što sam koristio/la StudioCode

Metodologija istraživanja

TAM (Davis, 1989.) uzima se kao model za procjenu percepcije i zadovoljstva učenika korištenjem tehnologije kao alata za učenje osnova računalstva u osnovnim školama. Kao i u većini istraživanja prihvaćanja tehnologije, u ovom se istraživanju uzima i pristup temeljen na teoriji te se postavljaju i testiraju odabrane hipoteze. Prikupljeni su podaci pomoću upitnika sa definiranim Likertovim ljestvicama sa mogućnošću ocjenjivanja od 1 do 5 za svaku nezavisnu varijablu.

Kao i kod većine objavljenih studija prihvaćanja tehnologije, za potrebe ovog istraživanja koristi se modeliranje strukturnih jednadžbi SEM za provjeru ispravnosti instrumenata i testiranje značajnosti između konstrukata. Kompletna analiza koja koristi SEM može se postići bilo kroz analizu kovarijancije (npr. LISREL, EQS i AMOS) ili parcijalnim najmanjim kvadratima PLS (engl. *Partial least squares*). Za odabranu PLS SEM analizu podataka korišten je softver SmartPLS3.

Hipoteze istraživanja

Na temelju tri čimbenika po TAM modelu, u nadograđenom modelu sa varijablom zadovoljstva za ciljani uzorak učenika, namjera korištenja web tehnologije za učenje osnova računalstva definirana je u smislu zadovoljstva učenika korištenjem tehnologije i njihovom odnosu, točnije stavovima prema tehnologiji. Razmatrane su pretpostavljene hipoteze za procjenu stava prema korištenju i zadovoljstva korištenja web aplikacije u procesu učenja među učenicima informatike u osnovnim školama.

Sudionici istraživanja

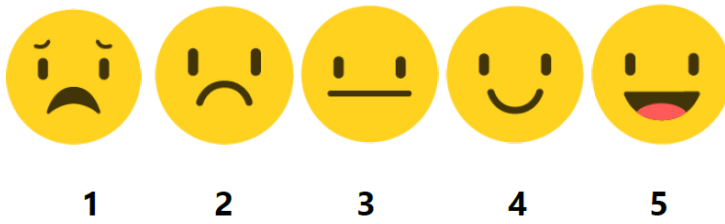
Uzorak od 114 učenika osnovnoškolaca koji su se željeli dodatno informatički educirati pohađajući Algebra Digitalnu akademiju, izvučeni iz nekoliko gradova diljem Hrvatske države (Zagreb, Osijek, Pula), odabrani su i provedeni kroz upitnik. Svi učenici odabrani u uzorku prošli su sve prethodno opisane aktivnosti unutar projektnih radionica sa korištenjem web aplikacije Code.org prilikom poučavanja osnova računalstva i kodiranja, te su imali prilagođenu obuku na web aplikaciji ovisno o dobi i prethodnom iskustvu osnova kodiranja zajedno sa općom uporabom računala.

Istraživački instrumenti i prikupljanje podataka

U istraživanju je korišten anketni upitnik, koji je osmišljen po teorijskim referencama prethodnih radova na ovom području uglavnom, Davis (1989.), Williams (2006.) i Napaporn (2007.). Likertova ljestvica bila je upotrijebljena za označavanje stupnja prihvaćanja gdje 1 označava da se uopće ne slažete sa navedenom tvrdnjom a 5 da se u potpunosti slažete sa navedenom tvrdnjom kao što je prikazano na slici 40.

Slika 39. Primjena Likertove skale u upitniku učeničke evaluacije web aplikacije

Moguće oznake:



- 1** - U **potpunosti** se **ne slažem** sa navedenim
- 2** - **Djelomično** se **ne slažem** sa navedenim
- 3** - **Niti** se slažem **niti** se ne slažem sa navedenim
- 4** - **Djelomično** se **slažem** sa navedenim
- 5** - U **potpunosti** se **slažem** sa navedenim

Izvor: autorov upitnik

Prikladan pristup uzorkovanja usvojen je kako bi se provjerile hipoteze. U školi je provedena anketa podijeljena svakom učeniku na papiru, a rješavanje je omogućeno flomasterom u boji.

Analiza podataka

Deskriptivna statistika

Ukupno je prikupljeno 114 važećih anketa (od 116 ispunjenih postojale su 2 ankete koje su odbačene jer su odgovori učenika odskakali od uzroka i smatrani su „outlier“-ima istraživanja) u periodu od 3. ožujka do 30. travnja 2018. Sudjelovanje u istraživanju je bilo dobrovoljno i učenici su obaviješteni da nema ispravnih ili krivih odgovora, i da njihovi će odgovori biti pohranjeni u povjerenju. Bilo je 83 učenika muškog spola i 31 ženskog, godišta kao što slijedi: 47 u dobi od 7. do 8. godine, 25 u dobi od 9. do 10. godina, 18 u dobi od 11. do 12. godine, 24 u dobi od 13. do 15. godina.

Model procjene uspješnosti

Istraživački model procijenjen je stupnjevito. Prvim mjerenjem se model analizirao nakon strukturne analize modela. Matrica faktor strukture opterećenja stavki i poprečnih opterećenja prikazana je u tablici 7. Konvergentna valjanost svakog konstrukta postiže se kao opterećenje predmeta za svaki konstrukt koji je iznad praga od 0,708 (Hair et al., 2014., 103). 6 stavki nije ispunilo minimalne zahtjeve i uklonjeni su iz modela.

Tablica 7. Matrica faktora opterećenja i poprečnih opterećenja faktora strukture

	ATT	PEOU	PU	SAT
ATT1	0.802	0.328	0.456	0.526
ATT2	0.808	0.167	0.513	0.493
PEOU1	0.320	0.811	0.344	0.190
PEOU3	0.174	0.804	0.429	0.213
PU1	0.539	0.391	0.865	0.475
PU3	0.452	0.368	0.731	0.369
PU4	0.453	0.394	0.808	0.328
SAT3	0.632	0.250	0.492	1.000

Izvor: Autorov izračun

Rezultati istraživanja

Konstrukcija i validacija instrumenata istraživanja

Analiza pouzdanosti koja je provedena za testiranje valjanosti i pouzdanosti konstrukata, dala je rezultate navedene u ovom poglavlju.

Za provjeru pouzdanosti prilikom korištenja Likertovih skala potrebno je izračunati i kreirati izvještaj "koeficijent pouzdanosti skale", tzv. *Cronbach-ov* alfa za unutarnju konzistenciju, te pouzdanost za mjerila ili pod skale koje se koriste. Analiza se vrši pod zajedničkim razmatranjem, a ne za analizu pojedinačne stavke. *Cronbach-ov* alfa može imati vrijednost u rasponu od 0 do 1, a unutarnja pouzdanost stavki u skali se kaže da je maksimalna kada je bliža 1. Vrijednost od 0.70 i više smatra se kriterijem za demonstraciju unutarnje dosljednosti ljestvice (Nunnally, 1978.). Dakle, teoretski, što je viši *Cronbach-ov* alfa, to je bolje, tj. korelacija između promatrane vrijednosti i istinske vrijednosti trebala bi biti što je moguće veća. Ovim radom su mjerena svih 14 pitanja upitnika, a ispitivanje pouzdanosti varijabli nakon uklanjanja nezadovoljavajućih

stavki otkrilo je *Cronbach*-ovu alfa višu od 0,786, kao što je prikazano u tablici 7, što potvrđuje pouzdanost ovih ljestvica unutar opće prihvaćenog raspona koji prelazi 0,70. Tablica 8. označava kompozitnu pouzdanost, *Cronbach*-ova alfa koeficijent, prosječno odstupanje odstupanja (AVE) zajedno s kvadratnim korijenima AVE (označeno brojevima u dijagonali) i korelacija između konstrukata pouzdanosti pokazatelja dobiveno je *Cronbach*-ovim alfa koeficijentom (Cronbach, 1970.), testirajući da je svaki pokazatelj sličan, kao i kompozitni koeficijent pouzdanosti (Werts et al., 1974.) koji je potrebno u obzir odgovarajuće pokazatelje. Konvergentna valjanost mjerena prosječnom varijacijom izdvojenog (AVE) predstavlja zajedničku varijancu između pokazatelja i njihov konstrukt i trebao bi biti veći od 0,5 (Hair et al., 2014, 103). Da bi potvrdili diskriminacijsku valjanost među konstrukcijama (Fornell-Lacker kriterij) AVE kvadratni korijen mora biti bolji od korelacije između konstrukata.

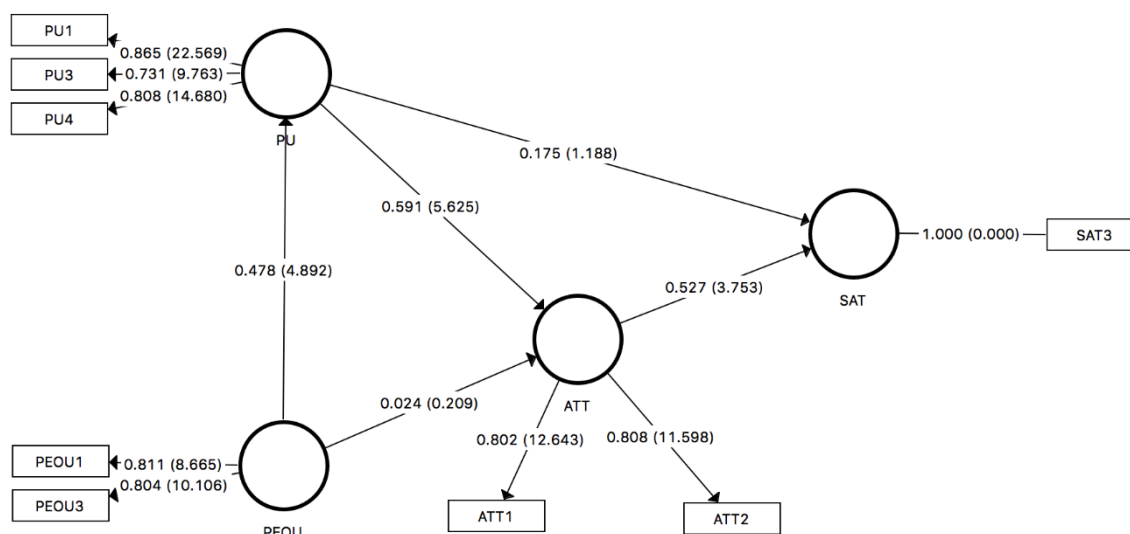
Tablica 8. *Cronbach*-ov alfa (CA), kompozitna pouzdanost (CR), prosječna varijacija izdvojenog (AVE) i diskriminacijska valjanost konstrukata (ATT, PEOU, PU, SATT)

	CA	rho_A	CR	AVE	ATT	PEOU	PU	SATT
ATT	0.457	0.457	0.787	0.648	0.805			
PEOU	0.466	0.466	0.789	0.652	0.307	0.807		
PU	0.722	0.733	0.844	0.645	0.602	0.478	0.803	
SAT	1.000	1.000	1.000	1.000	0.632	0.250	0.492	1.000

Izvor: Autorovi izračuni

Nakon utvrđivanja pouzdanosti za stavke i konvergentne i diskriminacijsku valjanosti konstrukata, ispitali smo strukturni model. Rezultati PLS analize za hipoteze H1 do H5 prikazani je na slici 2.

Slika 40. Rezultati strukturne analize podataka modela



Izvor: autorov PLS model

Model ne pokazuje probleme u kolinearnosti, kao prosječna inflacija blok varijancije je faktor VIF (AVIF) = 1,244 (prihvatljivo ako je ≤ 5) i prosječna puna ko-linearnost je VIF (AFVIF) = 1,346 (prihvatljivo ako je ≤ 5). Prosječni koeficijent puta (APC) je 0,359 ($p < 0,001$), dok je prosječni R-kvadrat (ARS) 0,337 ($p < 0,001$) i prosječno prilagođen R-kvadrat (AARS) je 0,328 ($p < 0,001$). Prediktivna sposobnost modela je tako zadovoljavajuća jer su svi R-kvadrati veći od 0,10 i oni se mogu tumačiti kao umjereni za stavove ($R^2 = 0,36$), nešto slabiji za percipiranu korisnost ($R^2 = 0,23$) i umjereni za zadovoljstvo ($R^2 = 0,42$).

Strukturni model pokazuje značajan pozitivan odnos između svih konstrukata osim odnosa između percipirane lakoće korištenja i stavova ka korištenju. Stoga su hipoteze H1, H2, H3, H4 i H5 podržane ($P < 0,01$). Na zadovoljstvo učenika izravno najviše utječu stavovi ka korištenju, a neizravno percipirana korisnost. R-kvadrat konstrukta percipirana korisnost je vrlo blizu umjerenom, ali je još uvijek slabije nego R-kvadrat konstrukta ATT i SAT. S povećanjem provedenih školskih aktivnosti korištenjem web aplikacije Code.org, mogao bi se dobiti bolji uvid. Ipak, rezultati pokazuju zadovoljstvo učenika Code.org platformom za učenje osnova računalstva i sadržaj koji trenutno pruža je suvremeni program za učenje u informatičkom osnovno školskom obrazovanju.

ZAKLJUČAK

U ovom radu kroz analizu istraživanja utvrđena je pouzdanost definiranih konstrukata koja pokazuje prikladnost pitanja koja su uključena u ovo istraživanje. Istraživanje potvrđuje da TAM kao korisni model pomaže istražitelju razumjeti stav učenika da koriste Code.org web tehnologiju u nastavi u kontekstu učenja osnova računalstva na satu informatike.

Rezultati pokazuju da učenička percepcija o korisnosti prilikom korištenja tog alata najviše utječe na njihov stav prema uporabi alata u nastavi. Percepcija učenika o jednostavnosti korištenja također je znatno utjecala na percepciju o korisnosti. Može se reći da će učenici smatrati IT alate korisnijim i imati će pozitivan stav prema integraciji tehnologije u nastavi informatike ako kroz odgovarajuću izobrazbu postanu stručniji u korištenju takvih alata.

Rezultati također ukazuju na potrebu proširenja istraživačkog modela kako bi se pronašli vanjski čimbenici / varijable kao što je model predložen poznatim TAM varijablama uz nadogradnju sa varijablom zadovoljstva korištenjem (SAT).

Na budućim istraživačima ostaje da trebaju ispitati nove varijable koje bi se mogle koristiti za proširenje modela TAM uz uključivanje nekih konstrukata vezanih uz konstrukte infrastrukture, tehnologije i subjekta koji mogu imati izravan ili neizravni (ali značajan) utjecaj na stavove učenika i namjeru korištenja računalne tehnologije. Ispitivanja prihvaćanja tehnologije bi se mogla provoditi pomoću TAM-a na različitim nastavnim alatima i kontekstima kako bi se istražila njegova valjanost. Potrebno je i dodatno ispitati kako se TAM može koristiti u predviđanju stvarne uporabe tehnologije. Kao i u svim empirijskim istraživanjima, ovo istraživanje također ima ograničenja kao što su: uzorak je ograničen na učenike Digitalne akademije u privatnoj obrazovnoj ustanovi Algebra koji se osposobljavaju za suvremene digitalne kompetencije, kako rezultati mogu varirati u slučaju drugih učenika u javnim školama, ti se rezultati ne mogu generalizirati; korišteni konstrukti temeljeni na izvornom TAM-u trebalo bi proširiti i uzeti u obzir sve konstrukte povezane s implementacijom IT-a koje mogu igrati glavne čimbenike u percepciji i stavovima učenika o uporabi informatičkih alata u nastavi. Iako istraživanje daje naznake da se TAM može koristiti za procjenu stava i može biti koristan za veliku populaciju učenika, generalizacija nalaza ograničena je samo na učenike informatike osnovnih škola.

Istraživanje je bilo uspješno i pokazalo je uzročne odnose između TAM varijabli (PU, PEOU i ATT) i nadograđene varijable zadovoljstva (SAT), te da ima dokaze koji podupiru primjenjivost TAM modela u pojašnjavanju stavova i zadovoljstva učenika informatike na Digitalnoj akademiji u gradovima diljem Hrvatske. Rezultati dobiveni iz ovog rada mogu pružiti korisni uvid u proces implementacije suvremenih ICT tehnologija primijenjenih u nastavi i razvoju e-sadržaja za informatiku u školama u Republici Hrvatskoj.

LITERATURA

JURČIĆ, M. (2007.), *Uravnotežena kombinacija socijalnih oblika nastave u funkciji oslobađanja potencijala učenja i poučavanja*, Kratis: Hrvatsko pedagogijsko društvo

BULJUBAŠIĆ-KUZMANOVIĆ, V. (2007.), *Kompetencije i kompetentnost učitelja*, Osijek: Učiteljski fakultet

LUČIĆ, K. i MATIJEVIĆ, M. (2004.), *Nastava u kombiniranim odjelima*, Zagreb: Školska knjiga

BOGNAR, L. i Matijević, M. (2005.), *Didaktika*, Zagreb: Školska knjiga

DWIVEDI, Y. K., WADE M. R. i SCHNEBERGER S. L. (2012.), *Information Systems Theory, Explaining and Predicting, Our Digital Society*, Vol. 1

HAIR, J. F. Jr., et al., *Multivariate Data Analysis*, 7th Edition

HAIR, J. F. Jr. (2006.), *GBC Preconference Readings*, Harvard Business School Publishing Corporation

DAVIS, F. D. (1989.), "Perceived usefulness, perceived ease of use, and user acceptance of information technology", *MIS Quarterly*, **13** (3): 319–340, [doi:10.2307/249008](https://doi.org/10.2307/249008)

NUNNALLY, J. C. (1978). *Psychometric theory* (2nd ed.), New York: McGraw-Hill

WERTS, C. E., LINN, R. L. i JORESKOG, K.G. (1974.), *Intra class reliability estimates: Testing structural assumptions*. Educational and Psychological Measurement, 34, 25–33.

WILLIAMS, D. (2006). Groups and Goblins: *The Social and Civic Impact of Online Gaming*. *Journal of Broadcasting and Electronic Media*. 50(4), p. 651-670. Journal site.

ETINGER, D., ŠEHANOVIĆ, J. i RIBIĆ, A., *Measuring the success of e-library implementation: students perceptions and use*, The impact of migration on Coatian cultural diversity

NAPAPORN K. (2007.), *Examining a Technology Acceptance Model of Internet Usage by Academics within Thai Business Schools*, Victoria University Melbourne, Australia

Code.org 2017. godišnji izvještaj, <https://code.org/about/2017>

(Pristupljeno:15.04.2018.)

PARSONS, D. i HADEN, P., (2006.) *Parson's programming puzzles: a fun and effective learning tool for first programming courses*,
<https://dl.acm.org/citation.cfm?id=1151890> (Pristupljeno: 15.04.2018.)

POPIS SLIKA

Slika 1 Primjer zagonetke sa raznim težinama zadatka	8
Slika 2 Autorova upravljačka ploča i napredak učenika	9
Slika 3 Statistika učenika koji su rješavali Tečaj 1	10
Slika 4 Igra Školice na ploči.....	16
Slika 5 Upoznavanje sa sučeljem Code.org tečaja	17
Slika 6 Upute kako pokazati Javascript kod u Code.org	18
Slika 7. Zadani kod (složene blokove) koje treba ispraviti.....	19
Slika 8. Kviz pitanje višestruki izbor	21
Slika 9. Code.org: Tečaj 1, Nivo 9, zadatak 1.....	22
Slika 10. Kodiranje strip priče, rad učenika 1.razreda	24
Slika 11. Grafički prikaz stupnjeva	25
Slika 12. Zadatak spajanje parova i rješenje.....	30
Slika 13. Korištenje petlje „repeat until“	31
Slika 14. Zadatak i rješenje spajanja parova Labirinta	31
Slika 15. Višestruki točan izbor - kviz pitanje	32
Slika 16. Zadatak korištenja funkcija.....	32
Slika 17. Prikaz nove get funkcije	32
Slika 18. Prikaz "collect purple nectar" funkcije.....	33
Slika 19. Prikaz nove funkcije	33
Slika 20. Prikaz rješenja zadatka koristeći if-else pelju	34
Slika 21. Funkcija kojom ispitujemo koliko koji cvijet ima nektara	35
Slika 22. Prikaz rješenja 8. zadatka	35
Slika 23. Funkcija „pokupi nektar ili napravi med“	35
Slika 24. Kviz ispitivanje uvjeta.....	36
Slika 25. Ugnježdena petlja	36
Slika 26. Prikaz rješenja korištenjem while petlje i uvjeta.....	37
Slika 27. prikaz rješenja korištenjem svih naučenih petlji i uvjeta.....	37
Slika 28. Kviz pitanje ispravljanje greški u funkcijama.....	38
Slika 29. Zadatak crtanja simetričnog cvijeta uz slaganje blokova kodova.....	39
Slika 30. Primjer optimalnog rješenja 10. zadatka	40
Slika 31. Rješenja zadatka crtanja robota slaganjem kodnih blokova	41
Slika 32. Funkcija "draw a square"	42
Slika 33. Funcije za stvaranje zupčanika	43
Slika 34. Prezentacijski slajd radova učenikana temu Zupčanik	44
Slika 35. Rješavanje zadatka koristeći repeat-untill petlju.....	45
Slika 36. Rješenje zadatka koristeći petlje i uvjete.....	46
Slika 37. Zadatak kviz pitanje za ispitivanje uvjeta u labirinitu.....	46
Slika 38. Autorov prošireni TAM model.....	49
Slika 39. Primjena Likertove skale u upitniku učeničke evaluacije web aplikacije	52
Slika 40. Rezultati strukturne analize podataka modela.....	55

POPIS TABLICA

Tablica 1. Code.org ciljevi i postignuća.....	2
Tablica 2. Prikaz CS osnovnih tečajeva i njihova namjena i koncepti.....	4
Tablica 3. Prikaz CS koncepata iz programiranja u Code.org.....	6
Tablica 4. Prikaz težine i opis koncepta na pripadajućoj težinskoj razini.....	7
Tablica 5. Prikaz temeljna područja težine i opis težinske razine	12
Tablica 6. Popis likova, pozadina, brzine kretanja i raspoloženja.....	23
Tablica 7. Matrica faktora opterećenja i poprečnih opterećenja faktora strukture.....	53
Tablica 8. Cronbach-ov alfa (CA), kompozitna pouzdanost (CR), prosječna varijacija izdvojenog(AVE) i diskriminacijska valjanost konstrukata (ATT, PEOU, PU, SATT).....	54

PRILOG

Evaluacijski listić

1.1 Procjena portala za učenje programiranja StudioCode.org

Moguće oznake:



1 2 3 4 5

- 1 - U potpunosti se ne slažem sa navedenim
- 2 - Djelomično se ne slažem sa navedenim
- 3 - Niti se slažem niti se ne slažem sa navedenim
- 4 - Djelomično se slažem sa navedenim
- 5 - U potpunosti se slažem sa navedenim

Korisnost:

StudioCode mi je bio koristan prilikom učenja programiranja igrica

StudioCode mi je olakšao da razumijem na koji način se programiraju igrice

Kada bih nastavio koristiti StudioCode sigurno bih naučio/la programirati igrice

Znanje koje sam dobio/la korištenjem StudioCode sigurno će mi pomoći kada budem dalje učio/la programirati igrice

Lakoća uporabe:

Bilo mi je jednostavno koristiti StudioCode

Razumio/la sam napisane upute za rješavanje zadataka na StudioCode

Jednostavno sam koristio/la blokove za slaganje koda

Sigurno bih jednostavno savladao/la sve slijedeće tečajeve na StudioCode

Stavovi:

Mislim da je dobro koristiti StudioCode

Mislim da je pametno koristiti StudioCode

Sviđa mi se ideja da nastavim koristiti StudioCode

Zadovoljstvo:

Dok sam koristio/la StudioCode bilo mi je zanimljivo

Bilo mi je zabavno koristiti blokove i doći do rješenja zadatka u StudioCode

Zadovoljan/na sam što sam koristio/la StudioCode

SAŽETAK

Danas su djeca okružena digitalnim tehnologijama i obrazovnim web aplikacijama koje nisu bile dostupne prije nekoliko godina. Kao informatički edukator i instruktor Digitalne akademije u obrazovnoj ustanovi Algebra Pula za ovo istraživanje izabrala sam Code.org web aplikaciju za učenje osnova računalstva. U prvom dijelu rada nakon opisa web aplikacije prikazani su primjeri provedenih aktivnosti projektne nastave informatike koja se izvodila tijekom školske godine 2017./2018. na Algebra Digitalnoj akademiji u OŠ Vidikovac.

Cilj ovog rada bio je prilagodba istraživačkog modela koji opisuje stavove ka budućoj upotrebi i zadovoljstvu korištenjem Code.org web aplikacije, temeljenog na modelu prihvaćanja tehnologije (TAM) kojeg je razvio Fred Davis. Upitnik za evaluaciju web aplikacije Code.org kreiran je kako bi ispitali stavove prihvaćanja učenika prema korištenju web aplikacije i njihovo zadovoljstvo korištenjem web aplikacije. Podaci 114 učenika osnovnih škola (8-14 godina) prikupljeni su u privatnoj školi Algebra Digitalna akademija u nekoliko gradova Republike Hrvatske. Model se sastoji od četiri varijable: stavovi prema upotrebi, zadovoljstvo, percipirana korisnost i percipirana jednostavnost korištenja. Model je procijenjen korištenjem modela strukturne jednadžbe (SEM) za koji je korišten softver SmartPLS3. Rezultati pokazuju da učnička percepcija o korisnosti prilikom korištenja tog alata najviše utječe na njihov stav prema uporabi alata u nastavi. Rezultati dobiveni iz ovog rada mogu pružiti korisni uvid u proces implementacije suvremenih ICT tehnologija primijenjenih u nastavi i razvoju e-sadržaja za informatiku u školama Republike Hrvatske.

Ključne riječi : code.org web aplikacija, osnove računalstva, TAM model, PLS-SEM analiza, učnička evaluacija

ABSTRACT

Today, children are surrounded by digital technologies and educational web applications that were not available a few years ago. As an IT educator and instructor of the Digital Academy at the educational institution Algebra Pula for this research, I chose the Code.org web application for learning the basics of computing. In the first part of the work, after the description of the web application, examples of the implemented activities of project teaching of informatics, which were performed during the school year 2017. / 2018., are shown. at the Algebra Digital Academy in the OŠ Vidikovac Elementary School.

The aim of this paper was to adapt the research model that describes attitudes towards future use and satisfaction using the Code.org web application, based on the TAM model developed by Fred Davis. The Code.org evaluation questionnaire was created to examine the attitudes of students accepting the use of the web application and their satisfaction with the use of the web application. Data were collected from 114 elementary school students (8-14 years old) in the private school Algebra Digital Academy in several cities of the Republic of Croatia. The model consists of four variables: attitude towards usage, satisfaction, perceived usefulness and perceived ease of use. The model was estimated using the structural equation model (SEM) for which the SmartPLS3 software was used. The results show that the pupil's perceptions of usefulness when using this tool most affect their attitude towards the use of teaching tools. The results obtained from this paper can provide a useful insight into the process of implementing modern ICT technologies applied in the teaching and development of e-content for IT in schools in the Republic of Croatia.

Keywords: code.org web application, computer science fundamentals, TAM model, PLS-SEM analysis, students evaluation