UDK 004.43 Scratch
004.43:371.3
Stručni rad

Dr. sc. Nevzudin Buzađija
Sanela Mirvić

PROGRAMSKI JEZIK SCRATCH U OSNOVNIM I SREDNJIM ŠKOLAMA ZA EDUKACIJU BUDUĆIH PROGRAMERA

Sažetak

U današnje vrijeme u osnovnim i srednjim školama ne prate se savremeni trendovi u predmetu informatika, posebno je to prisutno kod izučavanja nastavne cjeline programski jezici. Naime, još se zadržao programski jezik Qbasic ili Pascal koji na učenike djeluju demotivirajuće zbog složene sintakse i što učenici ne vide svrhu rješavanja matematskih problema. Zbog toga je potrebno da se uvedu vizuelni programski jezici kao što je Scratch koji je motivirajući za učenike, jer mogu rješavati razne probleme realnog tipa u vizuelnom okruženju bez ulaska u samu suštinu izvornog koda. Primjenom Scratch programskog jezika bi se razvila logika kod učenika u rješavanju realnog problema i stvarala ljubav prema programiranju. Ovo je značajno kad se uzme u obzir potražnja za mladim programerima i što im to može biti budući poziv. U provedenom istraživanju ovog rada će se vidjeti koliko su učenici bolje zainteresovani za programiranje primjenom Scatch programskog jezika u odnosu na dosadašnje programske jezike koji traže poznavanje sintakse istih.

Ključne riječi*: programski jezici, nova metodologija, Scatch, igrice, realni problemi, učenici, nastavnici.*

Uvod

Kako bi pratili razvoj tehnologije i zahtjeve svjetske ekonomije, mladi ljudi danas moraju vladati funkcionalnim znanjima i sposobnošću povezivanja različitih disciplina u nove kompetencije. STEM (prirodne nauke, informatika, inžinjerstvo i matematika) princip će zamijeniti tradicionalni pristup učenju baziran na memorisanju i reprodukciji činjenica, u korist praktične provjere i primjene znanja tokom i nakon školovanja, kako bi mlade generacije pripremio za ekonomiju zasnovanu na znanju i učinio ih konkurentim na tržištu rada u budućnosti. Ovo je princip koji se danas počinje primjenjivati u mnogim obrazovnim sistemima u svijetu, a ulažu se napori da ovaj koncept bude prihvaćen i kod nas, te da se omogući uvođenje STEM kompetencija u sistem obrazovanja u BiH.

Predmet informatika, kao obavezan, se izučava od šestog do devetog razreda primarnog obrazovanja, po jedan čas sedmično. Učenici u sedmom razredu uče o programskim jezicima, najčešće QBasic-u i Pascal-u, te u većini škola više teorijski nego praktični dio. Što se tiče sekundarnog obrazovanja izučavanje predmeta informatika se odvija u zavisnosti od smjera koji učenici odaberu prilikom upisa škole. U mnogim školama smjerovi koji po NPiP uče programiranje biraju programski jezik Pascal. Broj časova namijenjen učenju programiranja je nedovoljan i najveći dio tih časova učenici uče sintaksu programskog jezika. Nastavnici preferiraju da daju zadatke iz oblasti matematike, što učenicima dodatno oteža programiranje i loše djeluje na njihovu motivaciju. Na osnovu saznanja sa područja Zeničko-dobojskog kantona možemo zaključiti da se ne pridaje veliki

značaj učenju programiranja, a sličnu situaciju imamo i u drugim školama na području BiH. Kod velikog broja nastavnika je još uvijek zastupljen tradicionalan pristup u radu, te izbjegavaju upotrebu savremenih sredstava uz pomoć kojih bi učenici na dosta lakši način trajno usvajali znanja iz različitih oblasti. Mali broj nastavnika posvećuje dovoljno pažnje oblasti programiranja, te ukazuju učenicima na značaj programiranja.

Da bi se motivacija učenika za učenje programiranja povećala moramo nastojati kroz redovnu nastavu učenike na što jednostavniji i zanimljiviji način uvesti u ovu oblast i time probuditi njihovu želju da proširuju svoja znanja samostalno ili kroz izvannastavne aktivnosti. Programski jezik Scratch, koji je tema ovog rada, bi mogao biti veoma dobar uvod za učenike u kreativno kodiranje. Ovaj jezik je prilagođen učenicima već od predškolske dobi, a temelji se na grafičkom povezivanju prethodno nacrtanih blokova (kao LEGO kockice). Pomoću njega učenici bi mogli raditi na raznim projektima i učiti širok spektar aktivnosti iz različitih predmeta, te na taj način proširivati svoje znanje.

Programiranje i apstraktno mišljenje

Sposobnost apstraktnog mišljenja se sporo razvija i potrebno je stalno vježbati. Zbog toga je veoma bitno učiti djecu programiranje od najranije dobi. Većina djece nema razvijeno apstraktno mišljenje zbog čega im je programiranje, koje je samo po sebi apstraktno, teško za razumijevanje. Rješenje za ovaj problem daju nam vizuelni programski jezici. Oni omogućavaju učenje programiranja u konkretnom okruženju uklanjajući probleme teške sintakse. Djeci i odraslima je prirodno razmišljati od konkretnog prema apstraktnom. Apstraktni pojmovi poput varijabli, petlji i sl. u vizuelnom okruženju pružaju konkretno iskustvo. Iako vizuelni jezici ne proučavaju sintaksu neophodnu u većini drugih programskih jezika, oni pružaju čvrstu osnovu za shvatanje principa programiranja.

Vizuelni programski jezik kao što je Scratch daje učenicima mogućnost kreiranja interaktivnih priča, igrica, animacija i projekata koristeći se naredbama koje su napravljene u obliku slagalica, tako da je vizuelno jasno koje se naredbe mogu složiti. Prema tome pažnja učenika je fokusirana na rješavanje problema gdje učenici mogu programirati u kontekstu realni/stvarni svijet.

Dosadašnja istraživanja

Naglim razvojem tehnologije porasla je i potreba za stalnim usavršavanjem i usvajanjem novih vještina, stoga je vrlo važno naučiti programirati kako bi pojedinac opstao u digitalnom svijetu (Rees, G. et all., 2017). Učenje programiranja i razvoj računarskog razmišljanja, vještine su koje pridonose procesu učenja kod djece i pomažu im da se suoče sa stvarnim životnim situacijama (Rees, G. et all., 2017). Passey, D. (2017) ističe šest glavnih razloga zbog kojih bi se škole i nastavnici trebali fokusirati na nastavu programiranja unutar obaveznog obrazovanja: promjene u ekonomiji, razvoj organizacijskih vještina, širenje zajednice koja sve više koristi tehnologiju kao sredstvo komunikacije, cjeloživotno učenje, učenje novih vještina i kompetencija koje zahtijeva razvoj tehnologije te interes učenika. Većina zemalja prepoznala je važnost digitalnog opismenjavanja od najranije dobi te u svoje kurikulume implementirala (ili planira) sadržaje za razvoj digitalnih kompetencija, no i dalje smo svjedoci brojnih izazova i prepreka uspješne integracije tih sadržaja u obavezno školovanje (Stefania B. et all., 2016).

Estonija je jedna od zemalja koja veliki značaj pridaje učenju programiranja zbog čega je 2012. godine pokrenula projekat u sklopu kojeg se vršila edukacija nastavnika iz oblasti programiranja, nakon čega su obučeni nastavnici uključili programiranje u svoje kurikulume čak od prvog razreda primarnog obrazovanja.

Finska i Belgija su imale planove za integraciju programiranja u školske kurikulume. Finska je svoje planove ostvarila 2016. godine, te je postala jedna od prvih evropskih zemalja koja je uvela programiranje i algoritamsko razmišljanje kao obaveznu aktivnost kroz sve nastavne predmete od prvog razreda osnovne škole. Slično Finskoj, Švedska 2018. godine uvodi programiranje kroz nastavne predmete poput matematike, tehnologije i društvenih nauka za učenike od prvog do devetog razreda osnovne škole. Učenici se u prvom i drugom razredu upoznaju sa algoritmima i načinom pisanja algoritama, dok u trećem razredu počinju raditi u vizuelnom programskom okruženju.

Također, programiranje je dio i novog nacionalnog kurikuluma za osnovne škole u Turskoj, kojim se predlaže programiranje pomoću blokova, tekstualno programiranje i robotika (Stefania B. et all., 2016).

Kako programiranje nije popularno među učenicima, odabir programskog jezika može biti ključan faktor za motivaciju i stav učenika prema programiranju. Brojna istraživanja su pokazala kako se programiranjem igara povećava motivacija učenika te da se na taj način mogu usvojiti osnovni koncepti programiranja.

Prvo, veće istraživanje na temu upotrebe vizuelno-blokovskog programskog jezika Scratch je provedeno u klubu pokraj Los Angelesa. U njemu su sudjelovali učenici od 8 do 18 godina, a proučavalo se korištenje Scratcha za izradu igara, animacija, videa, itd. Rezultati su pokazali kako su učenici savladali osnovne programske koncepte u Scratchu, kojeg su doživjeli zabavnim za korištenje (Maloney J. H. et all., 2008). Mnoga istraživanja su pokazala da vizuelni jezici mogu utjecati na afektivnu i kognitivnu domenu učenika.

Istraživanje provedeno u Škotskoj, u školskom okruženju, pokazalo je kako su učenici programiranjem u Scratchu kreirali igre korištenjem programskih koncepata (Amanda W., et all., 2013).

Metodologija istraživanja

Predmet istraživanja

Predmet ovog istraživanja je proučavanje pristupa učenju programiranja kroz vizuelni programski jezik Scratch u osnovnoj i srednjoj školi. Poučavanje i učenje programiranja je izuzetno zahtjevno, posebno za početnike jer zahtjeva visoku razinu apstrakcije. Način poučavanja programiranja uglavnom se temelji na rješavanju matematičkih problema u tekstualnim programskim jezicima što dodatno smanjuje motivaciju učenika za učenjem programiranja.

Ciljevi i zadaci istraživanja

Ciljevi ovog istraživanja su:

- Organizovati nastavu zasnovanu na korištenju Scratch programa;
- Upoznati učenike sa Scratch razvojnim okruženjem;
- Predstaviti učenicima kroz različite igre primjenu Scratch programa u drugim predmetima, koristeći pri tome aktivni nastavni plan i program;

- Motivisati učenike da samostalno osmisle i kreiraju igricu u Scratch programu;

Na osnovu definisanog problema, predmeta i ciljeva istraživanja proizilaze sljedeći zadaci:

- Ispitati motivisanost učenika, prije i poslije eksperimentalnog programa, o učenju programiranja;
- Ispitati mišljenje učenika o kreiranju aplikacija u Scratch programu;
- Utvrditi da li je učenje Scratch programa pomoglo učenicima da bolje razumiju osnovne koncepte programiranja, te lakše usvoje znanja iz nastavnih predmeta, koja su im izložena kroz različite aplikacije kreirane u Scratch programu;

Hipoteze

Na osnovu istraživačkih pitanja postavljena je sljedeća hipoteza:

H_0: Primjena Scratch platforme u nastavi utiče na motivaciju učenika da se bave razvojem aplikacija i da postižu bolje rezultate učenja korištenjem aplikacija kreiranih u Scratch programu.

Uzorak istraživanja

Istraživanje je provedeno u prirodnom, školskom okruženju, što je primjereno za ovu vrstu istraživanja, jer se time eliminiše neprirodno ponašanje učenika.

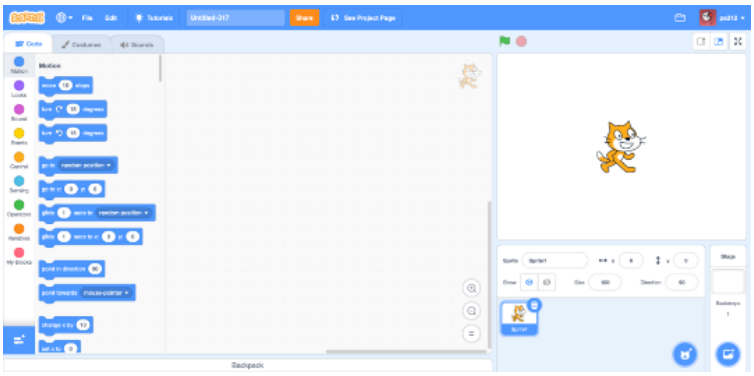
U istraživanju su učestvovali učenici iz dvije škole, jedna srednja i jedna osnovna škola. Iz obe škole učestvovala su po dva odjeljenja. Iz SŠ učestvovalo je ukupno 25 učenika prvog razreda, dok su iz OŠ u istraživanju učestvovala dva odjeljenja devetih razreda sa ukupno 35 učenika.

Programski jezik Scratch

Scratch je programski jezik otvorenog koda razvijen 2003. godine na Tehnološkom institutu države Massachusetts (Massachusetts Institute of Technology – MIT) kao projekt Lifelong Kindergarten grupe. Naredbe su napravljene u obliku slagalica i grupisane tematski sa različitim oblicima i bojama, tako da je vizuelno jasno koje se naredbe mogu složiti zbog čega nije potrebno prethodno znanje iz programiranja.

Osnovni elementi Scratch projekta

Scratch razvojno okruženje nalazi se na jednom prozoru i elementi su fiksni s više pojedinih dijelova. Na lijevoj strani prozora se nalaze pomoćne opcije i paleta blokova, u sredini je prostor za slaganje skripti, a na desnoj strani postoji mjesto gdje se program izvodi ispod čega se nalazi popis svih objekata u programu, njihove osobine i pozicije.



Slika 1. *Okruženje Scratch programa*

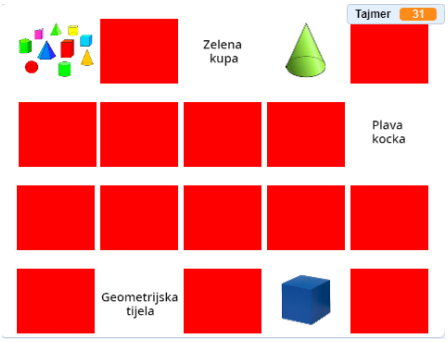
Osnovu projekta u Scratch-u čine objekti, zvani likovi (sprajtovi). Osnovni lik u Scratch projektu je mačka. Izgled lika se može izmijeniti dodajući mu drugi kostim. Lik može izgledati poput osobe, životinje ili bilo čega drugog. Moguće je dodati više likova crtanjem u Paint Editor-u, biranjem lika iznenađenja, učitavanjem sa računara ili metodom uhvati-prevuci-pusti preuzeti sa neke web stranice. Likovima u Scratch-u je moguće davati različite instrukcije poput kretanja, promjene boje, puštanja muzike, reakcije na druge likove ili interakcije sa scenom. Svaki lik ima vlastito ponašanje opisano naredbama (blokovima). Za davanje ovakvih instrukcija potrebno je složiti grafičke blokove u cjeline nazvane skripte. Skripte za više likova mogu se paralelno izvoditi.

Primjena Scratch-a u osnovnoj i srednjoj školi

Programski jezik Scratch, osim što se smatra idealnim za početke učenja programiranja, nudi priliku korištenja računara u svakodnevnim školskim aktivnostima iz različitih predmeta. Prezentovanje novog gradiva i ponavljanje naučenog uz pomoć priča, animacija i edukativnih igara kreiranih u Scratch-u čini nastavu zanimljivijom i jednostavnijom za učenike i pri tome oni trajno usvajaju nastavno gradivo koje im je prezentovano na ovaj način.

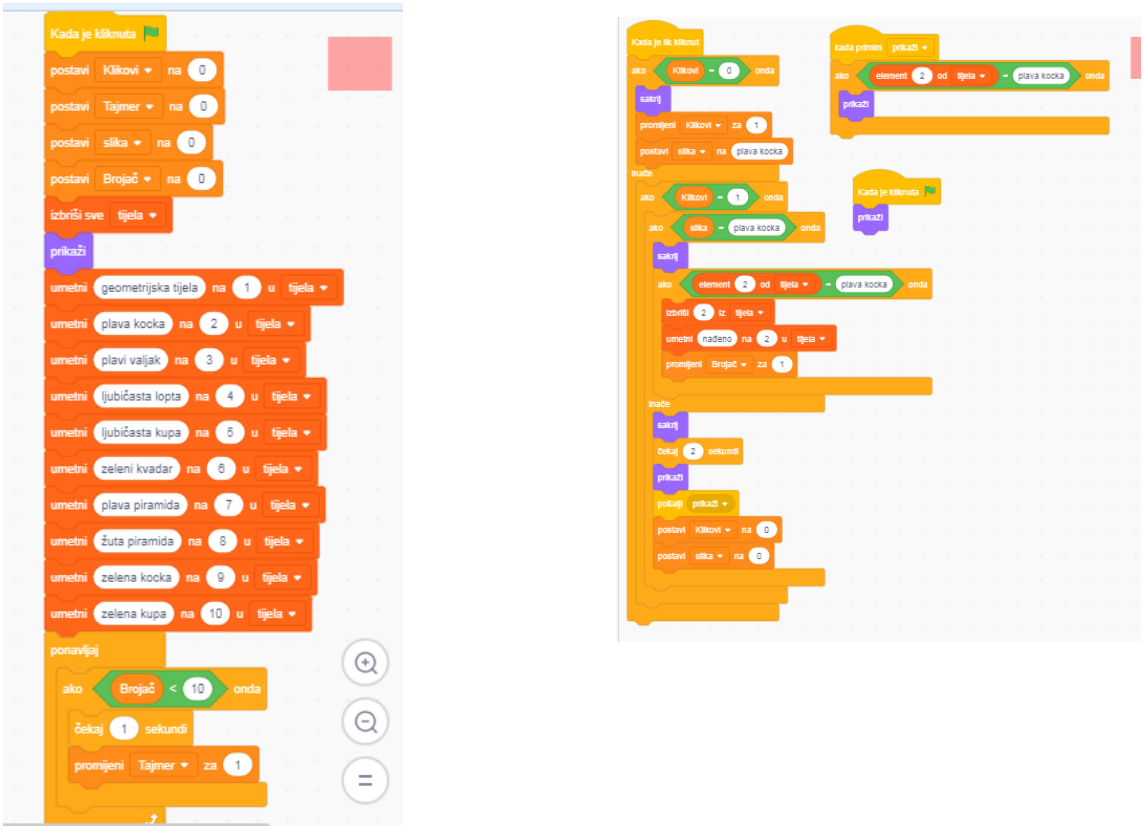
Jedna od igrica koju su osmislili i kreirali učenici 2. razreda srednje škole je „Memory igra“ sa geometrijskim tijelima. Njihova ideja je bila da iskoriste gradivo koje su učili iz predmeta matematika kako bi napravili zanimljivu igricu pomoću koje će ponoviti i utvrditi to gradivo, a koja im služi za razvoj pamćenja i koncentracije.

Igra se sastoji od 10 parova sličica koje je potrebno pronaći. Jedan par čini slika geometrijskog tijela koje su učenici preuzeli sa interneta i naziv tog tijela koji su sami napisali kao lik. Nakon što kliknemo na jednu od karti prikazanih na slici 2. ona nestane sa pozornice, a prikaže se lik koji se nalazi iza nje. Ukoliko druga karta na koju kliknemo nije odgovarajuća, nakon dvije sekunde obje karte se ponovo prikažu na pozornici i prekriju likove koji se nalaze iza njih. Ako je druga karta na koju smo kliknuli par prvoj otkrivenoj karti onda one ostaju otkrivene do kraja igre.



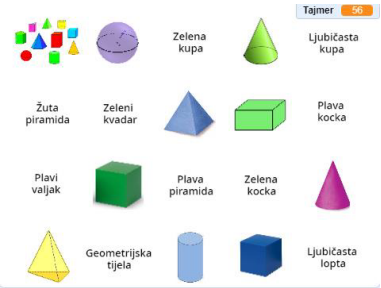
Slika 2. *Izgled pozornice tokom igre u "Memory igri"*

Kako bi postigli da pronađeni par likova ostane otkriven učenici su kreirali listu u koju se pokretanjem igre na zelenu zastavu upisuju nazivi svih geometrijskih likova koji se kriju iza karti. Nakon što se otkrije jedan par karti taj lik se briše iz liste, a na njegovu poziciju u listi se upisuje riječ „nađeno“. U nastavku igre karte prekrivaju samo likove čiji se nazivi nalaze na odgovarajućim pozicijama u listi. Za svaki od likova su kreirali lik „Karta“ koji ga prekriva, a skripte za parove su iste. Skripta koja se izvršava nakon pokretanja igre na zelenu zastavu je prikazana na sljedećoj slici, a napisana kod jednog od likova „Karta“.



Slika 3. *Skripta za lik "Karta"*

U igri je kreirana i varijabla „Tajmer“ čija se vrijednost na početku igre postavlja na nulu, a tokom igre mjeri vrijeme u sekundama. Nakon što sve parove likova otkrijemo dobijemo izgled pozornice kao na slici 4.



Slika 4. *Kraj "Memory igre"*

Mogućnost kreiranja projekata u Scratch-u koji im mogu poslužiti za učenje i zabavu su učenike motivisali i zainteresovali za učenje programiranja koje im se prije ovog istraživanja činilo teškim. Učenicima se svidjela i mogućnost što u svakom trenutku mogu pokrenuti kreirane skripte, te ih po potrebi ispravljati ili nadopunjavati određenim blokovima razvijajući svoje algoritamsko mišljenje.

Rezultati istraživanja

Kao što smo već naveli u istraživanju je učestvovalo ukupno 60 učenika, 35 učenika iz devetog razreda osnovne škole i 25 učenika iz srednje škole. Struktura uzorka učenika je prikazana u sljedećoj tabeli.

Istraživanje je bilo podijeljeno u nekoliko faza: pre-test znanja, ispunjavanje ulazne ankete, demonstracija razvoja programa u okruženju programskog jezika Scratch, samostalan rad učenika, provjera usvojenosti osnovnih koncepata (post-test) i izlazna anketa.

Pre-test se sastojao od tri zadataka u kojima su učenici trebali dijagramom toka ili pseudokodom predstaviti rješenje datog problema. U prvom zadatku je bilo potrebno napisati linijsku, u drugom razgranatu, a u trećem cikličnu algoritamsku strukturu. Svi zadaci su nosili po jedan bod, tako da su učenici mogli na pre-testu osvojiti najviše 3 boda, a najmanje 0 bodova.

Tabela 1. *Frekvencija tačnih odgovora na pre-testu*

Pre-test	**Frekvencija odgovora**			
	Osnovna škola		Srednja škola	
	Tačno	Netačno	Tačno	Netačno
Zadatak 1	19	16	20	5
Zadatak 2	0	35	4	21
Zadatak 3	0	35	2	23

Iz tabele 1. vidimo da su učenici srednje škole pokazali malo bolje rezultate u odnosu na učenike osnovne škole od kojih nijedan učenik nije tačno uradio drugi i treći zadatak.

Nakon eksperimentalnog programa učenici su radili post-test čiji je cilj bio utvrditi kakvo je znanje učenika iz programskog jezika Scratch kao i nivo algoritamskog mišljenja postignut upotrebom vizuelnog programskog jezika. Post-test se sastojao od 5 zadataka, a svaki od njih je nosio 1 bod kao i na pre-testu. Frekvencija tačnih odgovora na post-testu je prikazana u tabeli 2.

Tabela 1. *Frekvencija tačnih odgovora na post-testu*

Zadaci (post-test)	**Frekvencija odgovora**			
	Osnovna škola		Srednja škola	
	Tačno	Netačno	Tačno	Netačno
Prepoznavanje naredbi	35	0	25	0
Predviđanje ishoda izvršavanja programa	27	8	23	2
Predviđanje ishoda izvršavanja programa - ponavljanje	21	14	23	2
Predviđanje ishoda izvršavanja programa - grananje	15	20	13	12
Pisanje algoritma	12	23	10	15

Svi učenici su tačno uradili prvi zadatak čime su pokazali da su uspješno savladali programski jezik Scratch, odnosno da razumiju naredbe napisane na blokovima. U preostala četiri zadatka su učenici srednje škole, koji su osvojili 75,2% od ukupnog broja bodova, pokazali bolje rezultate u odnosu na učenike osnovne škole koji su osvojili 62,86% od ukupnog broja bodova.

Možemo zaključiti da su učenici osnovne škole popravili svoje rezultate u odnosu na učenike srednje škole, te da se razlika u postotku tačno urađenih zadataka smanjila u odnosu na pre-test. Razlika u osvojenim bodovima na post-testu nije statistički značajna potvrdio nam je i T-test koji nam je za isti nivo značaja (0,05) dao rezultate $t = 1{,}88,\ Sig(2-tailed) = 0{,}06 > 0{,}05$ (Tabela 4.). Prema tome ne možemo da tvrdimo da su učenici srednje škole u odnosu na učenike osnovne škole ostvarili bolje rezultate na post-testu kao što je to slučaj na pre-testu.

Tabela 3. *Rezultati T-testa za post-test dobijeni u programu SPSS (Group Statistics)*

	Škola u kojoj je rađeno istraživanje	N	Mean	Std. Deviation	Std. Error Mean
Suma rezultata dobijenih na post-testu	SŠ	25	3,76	1,09	0,22
	OŠ	35	3,14	1,35	0,23

Tabela 4. *Rezultati T-testa za post-test dobijeni u programu SPSS*

		F	Sig.	t	df	Sig. (2-tailed)	Mean	Std. Error	95 % Confidence Interval of the Difference	
									Lower	Upper
Suma rezultata dobijenih na post-testu	Equal variances assumed	3,00	0,09	1,88	58	0,06	0,62	0,33	-0,04	1,27
	Equal variances not assumed			1,95	57,08	0,06	0,62	0,32	-0,02	1,25

U tabeli 4. je prikazana frekvencija tačnih odgovora za svih 60 učenika na pre-testu i post-testu. Na pre-testu su učenici od ukupnog broja bodova osvojili samo 25 %, dok su na post-testu osvojili 68 %. Peti zadatak na post-testu je uradilo čak 36,67 % učenika, a na pre-testu algoritam grananja, drugi zadatak, je uradilo samo 6,67 % učenika. Iz toga možemo zaključiti da je upotreba programskog jezika Scratch doprinijela razvoju algoritamskog mišljenja kod učenika.

Tabela 4. *Frekvencija tačnih odgovora na pre-testu i post-testu*

Zadaci	**Frekvencija tačnih odgovora**			
	Pre-test		Post-test	
Zadatak 1	39	65%	60	100%
Zadatak 2	4	6,67%	50	83,33%
Zadatak 3	2	3,33%	44	73,33%
Zadatak 4	/	/	28	46,67%
Zadatak 5	/	/	22	36,67%

Upotrijebivši metode deskriptivne statistike na zbirne rezultate dobijene na pre-testu i post-testu (tabela 5) vidimo da se prosječan broj bodova po učeniku na pre-testu iznosio 0,75, dok je na post-testu svaki učenik u prosjeku osvojio 3,4 boda. Na pre-testu je 50 % učenika osvojilo 1 ili 0 bodova, dok je na post-testu 50 % učenika osvojilo 4 ili 5 bodova. Najveći broj učenika je na post-testu osvojio 2 boda, što je također, bolji rezultat u odnosu na pre-test gdje je najviše njih imalo osvojen 1 bod. Iz tabele 5 vidimo i da su svi učenici na post-testu uradili bar jedan zadatak tačno, a na oba testiranja smo imali učenike koji su osvojili maksimalan broj bodova.

Tabela 5. *Deskriptivna statistika za zbirne rezultate pre-testa i post-testa*

	N	**AS**	**Medijan**	**Mod**	**Min**	**Max**
Pre-test	60	0,75	1	1	0	3

Post-test	60	3,4	3,5	2	1	5

Kako bismo pokazali da su učenici značajno napredovali, odnosno da su ostvarili bolje rezultate na post-testu u odnosu na pre-test koristit ćemo F-test (test varijanse) sa nivou značaja od 0,05.

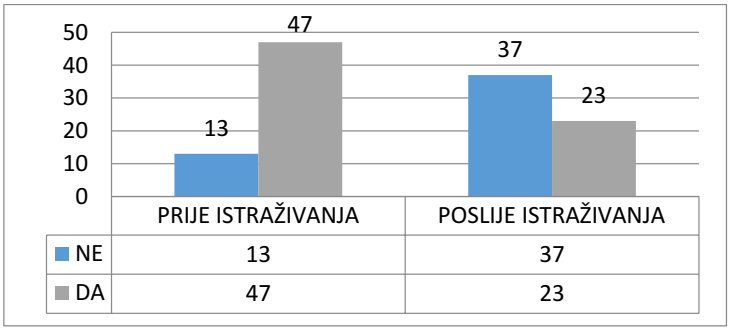
Tabela 6. *Rezultati F-testa dobijeni*

F-Test Two-Sample for Variances		
	Variable 1	*Variable 2*
Mean	3,4	0,75
Variance	1,634	0,462
Observations	60	60
Df	59	59
F	3,538	
p(F<=f) one-tail	1,40E-06	
F Critical one-tail	1,5399	

Iz tabele 6. vidimo da je za dobijeno F vrijednost $p < 0{,}05$ što znači da postoji statistički značajna razlika između varijansi koje su učenici ostvarili na pre-testu i post-testu. Prema tome možemo reći da su učenici upotrebom programskog jezika Scratch postigli bolje rezultate učenja i razvili algoritamski način razmišljanja, te se potvrdila hipoteza H0.

Učenici su prije i poslije eksperimentalnog programa popunjavali anketu zadovoljstva nastavom informatike i programiranjem.

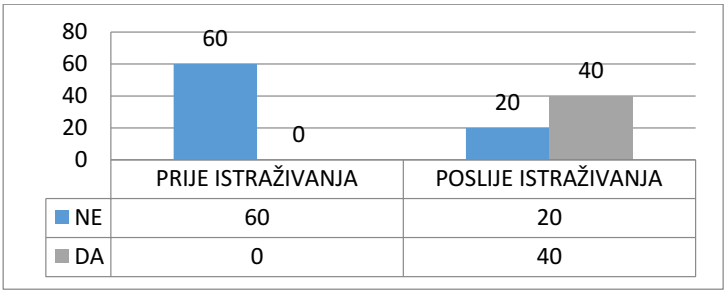
Na pitanje („Misliš li da je programiranje teško?“) je prije istraživanja 78,33 % učenika odgovorilo sa „DA“, dok je isti odgovor poslije istraživanja dalo 38,33 % učenika. Da se broj učenika koji smatraju da je programiranje teško smanjio nakon eksperimentalnog programa vidimo i na grafikonu 1.



Grafikon 1. *Frekvencija odgovora na pitanje „Misliš li da je programiranje teško?“*

Hi-kvadrat test ($\chi^2 = 56{,}563, df = 1, p = 0{,}000 < 0{,}05$) za ovo pitanje pokazuje da postoji statistički značajna razlika između rezultata koje smo dobili poslije eksperimentalnog programa i onih koje smo očekivali prema rezultatima prije eksperimentalnog programa.

Da niko od učenika nije, prije istraživanja, samostalno kreirao aplikaciju koristeći programske jezike potvrdili smo pitanjem u anketi („Da li si samostalno kreirao aplikaciju koristeći programske jezike?“). Poslije istraživanja je 40 učenika odgovorilo sa „DA“ na ovo pitanje (Grafikon 2), gdje je njih 23 navelo nazive igrica i kvizova koje su kreirali na času ili samostalno.



Grafikon 2. *Frekvencija odgovora na pitanje „Da li si samostalno kreirao aplikaciju koristeći programske jezike?“*

Wilcoxonov test ($Z = -6{,}325, p = 0{,}000 < 0{,}05$) za ovo pitanje nam potvrđuje da je razlika u odgovorima prije i poslije eksperimentalnog programa statistički značajna, te da se broj učenika koji su samostalno kreirali aplikaciju povećao.

Zaključak

Možemo zaključiti da učenici nakon provedenog eksperimentalnog programa imaju pozitivnije stavove po pitanju učenja programiranja na nastavi informatike, te da su motivisani za kreiranje aplikacija. Učenici su više motivisani za samostalno učenje programiranja i kreiranja vlastitih aplikacija. Istraživanje je potaknulo učenike i da aplikacije koriste na nastavi drugih predmeta što im je pomoglo da lakše usvajaju gradivo koje uče.

Učenje programiranja se mora prilagoditi net generaciji učenika koji preferiraju vizuelno okruženje i stvaranje rješenja za realne probleme kao što su igrice, kvizovi, upravljanje robotima itd. Ovo istraživanje pokazuje da se obrazovni sistem mora prilagoditi posebno u predmetu informatika prema stvaranju informatički pismenih mladih ljudi koji znaju stvarati novu vrijednost i koji će biti korisni članovi društva. Softverska industrija se progresivno razvija i sve su veći zahtjevi za vještinama programiranja koja su primjenjiva za rješavanje raznih problema. Scratch programski jezik bi trebalo uvesti u svim osnovnim i srednjim školama jer istraživanje pokazuje da učenici su motivisani i brže razvijaju logiku za rješavanje problema na računaru.

Literatura:

1. Amanda, W., Thomas, H., Thomas, C. (2013). Using Scratch with Primary School Children: An Evaluation of Games Constructed to Gauge Understanding of Programming Concepts, International Journal of Game-Based Learning 3:93-109, DOI: 10.4018/ijgbl.2013010107
2. Europska komisija (2019). Digitalne vještine za sve Europljane; preuzeto sa: https://digital-strategy.ec.europa.eu/en/library/digital-skills-all-europeans-brochure
3. Maloney, J., Mitchel, R., Natalie, R., Peppler, K., & Kafai, Y. (2008). Digital Media Designs with Scratch: What Urban Youth Can Learn about Programming in a Computer Clubhouse. In Kanselaar, G., Jonker, V., Kirschner, P. A., & Prins, F. J. (Eds.), International Perspectives in the Learning Sciences: Cre8ing a learning world. Proceedings of the Eighth International Conference for the Learning Sc-iences – ICLS 2008, Volumes 3 (pp. 81-82). Utrecht, The Netherlands: International Society of the Learning Sciences.
4. Mark, N., Aidan, M. (2019). *Visual and textual programming languages: a systematic review of the literature*; https://core.ac.uk/download/pdf/297032078.pdf
5. Ministarstvo za obrazovanje i istraživanje Estonija; *Nacionalni kurikulumi 2014;* https://www.hm.ee/en/national-curricula-2014

6. Passey, D. (2017). Computer Science (CS) in the Compulsory Education Curriculum: Implications for Future Research Education and Information Technologies, v22 n2 p421-443 Mar 2017
7. Patrik, L. (2017.) Programski jezici namijenjeni učenju programiranja; URL: https://bib.irb.hr/datoteka/892580.zavrsni_rad_pl.pdf
8. Rees, G., Tonon, G., Mikkelsen, C., & de la Vega, L. R. (2017). Urban-rural variations in children's lives and subjective well-being: A comparative analysis of four countries. Children and Youth Services Review, 80, 41-51.
9. Stefania, B. et. all. (2016). Developing Computational Thinking: Approaches and Orientations in K-12 Education, Conference: EdMedia 2016, Vancouver

SCRATCH PROGRAMMING LANGUAGE IN PRIMARY AND SECONDARY SCHOOLS FOR THE EDUCATION OF FUTURE PROGRAMMERS

Abstract

Nowadays, modern trends in computer science are not followed by primary and secondary schools, this is especially present in the study of teaching unit programming languages. Namely, the Qbasic or Pascal programming language has been retained, which has a demotivating effect on students because of its complex syntax and the fact that students do not see the purpose of solving mathematical problems. So, it is necessary to introduce visual programming languages such as Scratch, which is motivating for students, because they can solve various real-type problems in the visual environment without going into the very essence of the source code. Applying the Scratch programming language would develop students logic in solving a real problem and it would create a love for programming. This is significant when you take in consideration the demand for young programmers and what their future call may be. This research will show how much better students are interested in programming using the Scratch programming language compared to previous programming languages that require knowledge of their syntax.

Key words: *programming languages, new methodology, Scatch, games, real problems, students, teachers.*