

## Prihvatanje i korištenje Scratcha među početnicima u programiranju

ANA MARTINOVIĆ <sup>1</sup>, NIKOLA MARANGUNIĆ <sup>2</sup>

<sup>1</sup> Prirodoslovno-matematički fakultet Sveučilišta u Splitu, Ruđera Boškovića 33, Hrvatska,  
e-pošta: amartinov@pmfst.hr

<sup>2</sup> Prirodoslovno-matematički fakultet Sveučilišta u Splitu, Ruđera Boškovića 33, Hrvatska,  
e-pošta: nikola@pmfst.hr

\*Kontakt autor: nikola@pmfst.hr

**Sažetak** Računalno programiranje razvija vještine rješavanja problema i logičkog zaključivanja te je njegova integracija u sve obrazovne razine nužna. Međutim, učenje programiranja nije jednostavno i predstavlja izazov i za učenike i za nastavnike. Zbog svoje jednostavnosti *Scratch* se koristi kao uvod u programiranje, a računalne vještine koje učenici razvijaju u *Scratchu* mogu se kasnije primijeniti u drugim programskim jezicima. U ovom radu istraživala se povezanost četiriju varijabli modela TAM, percipirane korisnosti, percipirane lakoće korištenja, stava prema korištenju i namjere korištenja programskog jezika *Scratch* kako bi se otkrio stav i namjera stvarnog korištenja *Scratcha* među početnicima u programiranju. Prema kvantitativnim rezultatima može se jasno reći da su svi učenici voljeli programiranje i željeli su ga poboljšati. Rezultati su, između ostalog, pokazali i kako učenici smatraju *Scratch* korisnim i jednostavnim za korištenje, ali ipak nedovoljno ozbiljnim za učenje drugih tekstualnih programskih jezika.

**Ključne riječi:** model prihvatanja tehnologije (TAM); računalno razmišljanje; srednjoškolski učenici; strategije poučavanja; vizualno programiranje;

## Perceived acceptance and use of Scratch among beginners in programming

ANA MARTINOVIĆ <sup>1</sup>, NIKOLA MARANGUNIĆ <sup>2</sup>

<sup>1</sup> Faculty of Science, University of Split, Ruđera Boškovića 33, Croatia,  
e-mail: amartinov@pmfst.hr

<sup>2</sup> Faculty of Science, University of Split, Ruđera Boškovića 33, Croatia,  
e-mail: nikola@pmfst.hr

\* Correspondence: nikola@pmfst.hr

**Abstract** Computer programming develops problem-solving and logical reasoning skills, therefore its integration into all levels of education is essential. However, learning programming is not easy and poses a challenge to both students and teachers. Because of its simplicity, Scratch is used as an introduction to programming, and the computer skills that students develop in Scratch can later be applied in other textual based programming languages. In this research, the relationship between four variables of the TAM model - perceived usefulness, perceived ease of use, attitude towards use and intentions to use Scratch - were investigated to reveal the attitude towards and intention to use Scratch among beginners in programming. According to the quantitative results, it can be clearly said that all students loved programming and wanted to improve their programming. The findings showed, that students find Scratch easy to use and useful, but still not “serious” enough to prepare them for “real programming” in text based programming languages.

**Keywords:** Technology Acceptance Model (TAM), computational thinking, secondary school students, learning strategies, visual programming

## 1 Uvod

Živimo u digitalnom dobu i računala su dio naše svakodnevnice. Djeca tehnologiju prihvaćaju na „prirodan“ način, nemaju strah od nje i većinu slobodnog vremena provode uz tehnologiju. Vrijeme koje provode uz tehnologiju treba iskoristiti i usmjeriti na učenje. „Najbolji način za to je učenje programiranja od najranije dobi, ali na način koji je djeci „prirodan“ (Bubica, Mladenović i Boljat, 2013). Programirati znači komunicirati i upravljati računalom, stvarati vlastite sadržaje, rješavati probleme, ispravljati pogreške. Programirati danas znači biti pismen (Resnick et al., 2009).

Istraživanje u ovom radu provedeno je među učenicima prvih i drugih razreda četverogodišnje strukovne škole kojima je *Rješavanje problema pomoću računala jedna* od nastavnih cjelina u izvedbenom godišnjem kurikulumu. Prije uvoda u tekstualni programski jezik *Python*, implementiran je *Scratch* kako bi se učenici na jednostavan i zabavan način upoznali s osnovnim principima programiranja, razvojem algoritama i rješavanjem problema. Cilj istraživanja je izmjeriti prihvaćanje vizualnog programskog jezika u srednjoj školi analizirajući utječu li percipirana korisnost, stav prema korištenju i percipirana lakoća korištenja na namjere učenika za stvarnom upotrebom *Scratcha*. Istraživanje je utemeljeno na modelu prihvaćanja tehnologije (eng. *Technology Acceptance Model*, TAM). Istraživačka pitanja na kojima se temelji ovaj rad bila su:

- 1) U kojoj mjeri percipirana korisnost utječe na stav učenika o *Scratchu*?
- 2) U kojoj mjeri percipirana lakoća korištenja utječe na namjeru korištenja *Scratcha*?
- 3) U kojoj mjeri stav prema korištenju *Scratcha* utječe na namjeru korištenja *Scratcha*?

Programiranje je višeslojna vještina. Učenici moraju usvojiti algoritamski način rješavanja problema, razumjeti način na koji računalo radi, imati predodžbu o tome kako će se program izvršavati itd. Početnik u programiranju nema izgrađen model računala, odnosno ne poznaje princip rada računala i njegove mogućnosti (Ben-Ari, 1998a). Gomes i Mendes (2007) ističu kako problemi za mnoge učenike počinju već u ranoj fazi učenja kada trebaju riješiti konkretne probleme iz stvarnog svijeta na algoritamski način. Tan i suradnici su

zaključili da je moguć uzrok problema s učenjem programiranja nedostatak jasno izgrađenih mentalnih modela memorije računala. Većina studenata imala je problem s konceptima vezanim za memoriju kao što su pohrana i manipulacija varijablama u glavnoj memoriji računala (Tan, Yee i Ling, 2009). Milne i Rowe (2002) su zaključili da mnogi studenti ne mogu razviti jednostavan mentalni model tijeka izvođenja programa (Milne i Rowe, 2002).

Još jedan razlog poteškoća početnika u učenju programiranja je i sintaksa programskih jezika koja nije jednostavna. Mnoge pogreške i zablude u programiranju nastaju kao posljedica prijelaza s prirodnog na programerski jezik (Bonar i Soloway, 1983). Sintaksa programskog jezika je skup pravila i simbola koji programeru omogućuju pisanje ispravno strukturiranih programa. Često su mješavina engleskog jezika i različitih simbola, kao što su točke sa zarezom, zagrade i drugi simboli, čija upotreba zahtijeva „uvježbano oko“ pa mnogi početnici u programiranju budu obeshrabreni upravo krutom sintaksom.

U podučavanju programiranja često se koriste tradicionalni oblici nastave. Lekcije su fokusirane na dijelove programskog kôda i sintaksu programskog jezika. U procesu rješavanja zadataka koriste se dijelovi programskog kôda usvojeni kroz teorijski dio. Problem nastaje kod usvajanja i kombiniranja tih dijelova u smislenu cjelinu jer učenik kroz lekcije ne usvaja opće strategije primjene programiranja. On se više usredotočuje na rješavanje malih dijelova kôda, a ne na planiranje i testiranje cjelovitog rješenja. Isto tako, učenici ne predstavljaju homogenu skupinu s obzirom na sposobnosti i različite stilove učenja. Mnogi istraživači u području edukacije smatraju stilove učenja važnim faktorom procesa učenja. Učinkovitost u učenju je veća u učenika čiji je stil učenja u skladu s načinom predavljanja sadržaja koji uče.

Dio problema može ležati i u već usvojenom znanju učenika. Vrlo dobar primjer su simboli  $+$  i  $=$  koji imaju različita značenja u matematici i programiranju. Kognitivne poteškoće programera početnika stvara i brojanje od nule jer brojanje u stvarnom životu započinje od jedan i intuitivno je jasno da nulu ne treba brojiti. Za stvaranje dobrih temelja za razvoj ispravnih koncepata bitno je na vrijeme otkriti učeničke miskoncepcije (učenički koncept koji nije u skladu sa znanstvenom spoznajom) i suočiti ih s njima (Ben-Ari, 1998b).

Usvajanje vještine programiranja značajno ovisi i o motivaciji učenika. Programiranje zahtijeva dugotrajan rad, disciplinu i strpljivost. Svaki problem koji se rješava programiranjem, pokretanjem napisanog programa može raditi ili ne. Programiranje zahtijeva kontinuirano vježbanje, no ne mora značiti da će svi imati jednak uspjeh s obzirom na broj uloženi sati. Početnički neuspjeh može rezultirati padom motivacije i odustajanjem od daljnjeg ulaganja truda.

Snažan utjecaj u nastavi programiranja ima upotreba alata vizualizacije. Algoritmi i strukture podataka, kao osnovni građevni blokovi računalnih procesa i tehnika, su dinamični procesi koji se mijenjaju i statične slike ne mogu prenijeti ovu promjenu kao što mogu animirane slike. Općenito, možemo reći da vizualizacija olakšava učenje jer pomaže u stvaranju mentalnih modela apstraktnih pojmova. Upravo je iz tog razloga mnogo truda uloženo u razvoj alata za vizualno programiranje kojima je cilj kroz zabavno okruženje smanjiti tjeskobu i strah koji su tako često povezani s učenjem programiranja. Vizualno programiranje opisuje programsko okruženje u kojem su grafičkim ikonama, blokovima, predstavljeni elementi računalnog programa kao što su petlje, varijable, uvjetna grananja itd. Na taj način početnik može brzo napredovati u pisanju kôda bez učenja sintakse cijelog programskog jezika.

*Scratch* je jedan od najraširenijih vizualnih programskih jezika. Zbog svoje jednostavnosti koristi se kao uvod u programiranje, a računalne vještine koje učenici razvijaju u *Scratchu* mogu se kasnije primijeniti u drugim programskim jezicima poput *Pythona* i *Jave* (Wilson i Moffat, 2012; Armoni, Meerbaum-Salant i Ben-Ari, 2015). Primjenom *Scratcha* u edukaciji potiče se učenje temeljeno na iskustvu. Učenici uče, rješavaju probleme, međusobno surađuju, realiziraju svoje ideje te kritički razmišljaju. Te aktivnosti podrazumijevaju otvorenu komunikaciju, analizu i kontinuirano učenje.

*Scratch* je programsko okruženje koje omogućuje korisnicima, uglavnom u dobi između 8 i 16 godina, učenje i razumijevanje temeljnih koncepata programiranja kroz razvijanje svrhovitih i smislenih projekta poput igara ili animacija (Maloney, Resnick, Rusk, Silverman i Eastmond, 2010). Programiranjem u *Scratchu* korisnici mogu pozitivno razvijati kompetencije kao što su rješavanje problema i vještine kritičkog mišljenja, koje su ključne za 21. stoljeće (Marques

i Marques, 2012)problem solving and autonomy. Several tools can help to improve these competences (e.g. the SCRATCH programming language.

Armoni, Mareebaum-Salant i Ben-Ari (2015) razmatrali su prijelaz s učenja programiranja u vizualnom okruženju *Scratch* na učenje programiranja u profesionalnom tekstualnom programskom jeziku (C# ili Javi) u srednjoj školi. Utvrdili su da je programsko znanje i iskustvo učenika koji su naučili *Scratch* olakšalo učenje naprednijeg gradiva. Učenicima je bilo potrebno manje vremena za učenje novih tema, imali su manje poteškoća u učenju i bolje su razumjeli većinu koncepata. Ujedno, uočena je viša razina motivacije, samoučinkovitosti i interesa za računalstvo. Ovo istraživanje opravdava poučavanje programiranja općenito, a posebno vizualnog programiranja u srednjim školama (Armoni i dr., 2015).

*Scratch* je vrlo korisna okolina jer se programiranje izvodi izgradnjom blokova jednostavnih naredbi, a ne pisanjem tekstualnih naredbi. Theodorou i Kordaki (2010) dizajnirali su i razvili poznatu računalnu igricu *Super Mario* u okruženju *Scratch* kako bi srednjoškolcima približili temeljne koncepte u programiranju, poput pojma varijable (Theodorou i Kordaki, 2010). Učenje temeljeno na igricama, učenje temeljeno na problemu i vizualno programiranje mogu pomoći učenicima ostvariti bolje rezultate u početnom učenju programiranja. Razvoj igara uz programsko okruženje *Scratch* poboljšava i vještine programiranja i motivaciju učenika (Topalli i Cagiltay, 2018).

Vještina rješavanja problema uključuje shvaćanje problema, analizu problema, pronalaženje rješenja i pisanje programa, provjeru rješenja testiranjem i izmjenu programa prema rezultatu testa (Lai i Yang, 2011). Lai i Yang (2011) proveli su istraživanje kako bi procijenili učinak vizualnog programiranja na logičke sposobnosti učenika i vještine rješavanja problema. U istraživanju u kojem su sudjelovali učenici šestih razreda primijetili su značajno poboljšanje vještine rješavanja problema. Zaključeno je kako povezivanje vizualnog učenja programiranja i strategija rješavanja problema može poboljšati vještine rješavanja problema učenika.

U sličnom su istraživanju do malo drugačijih rezultata došli Jiang i Li (2021). Cilj ovog istraživanja bio je također analizirati učinke učenja jezika *Scratch* na vještine računalnog mišljenja (kreativnost, algoritamsko mišljenje,

kooperativnost, kritičko mišljenje i rješavanje problema) učenika osnovnih škola. Proveli su istraživanje s 336 učenika petih razreda. Rezultati istraživanja pokazuju kako postoji značajna razlika u vještinama kreativnosti, kooperativnosti i kritičkog mišljenja. Međutim, u ovom istraživanju *Scratch* nije postigao značajne razlike u vještinama rješavanja problema i algoritamskog razmišljanja. Učitelji bi trebali kombinirati *Scratch* s drugim predmetima, kao što su Matematika i Robotsko programiranje, kako bi učenicima pružili smislenije probleme programiranja koji razvijaju vještine algoritamskog razmišljanja (Jiang i Li, 2021) algorithmic thinking, cooperativity, critical thinking, and problem solving.

Wilson i Moffat (2010) bilježili su kognitivni napredak učenika i afektivni utjecaj svake lekcije tijekom 8 tjedana poučavanja u programskom okruženju *Scratch*. Istraživanje je provedeno s učenicima osnovnih škola u dobi od 8 do 9 godina. Rezultati su pokazali umjereni kognitivni napredak učenika i ugodnije iskustvo za učenike, čineći učenje programiranja pozitivnim iskustvom, suprotno frustraciji i tjeskobi, što tako često karakterizira uobičajeno iskustvo učenja (Wilson i Moffat, 2012).

Učenici percipiraju *Scratch* kao lagan i zabavan alat koji motivira učenje programiranja i potiče želju za upisivanjem studija programiranja (Permatasari, Yuana i Maryono, 2020).

*Scratch* je implementiran kao preliminarno programsko okruženje kako bi se poboljšalo iskustvo u programiranju, motivacija i uspjeh studenata računalnog inženjerstva (Arpaci, Durdu i Mutlu, 2019) motivation, and success in the Computer Science (CS. U fokusu istraživanja bilo je prihvaćanje i korištenje programskog okruženja *Scratch*. Istraživanje na uzorku od 186 studenata pokazalo je značajnu povezanost percipiranog uživanja s percipiranom korisnošću i stavovima. Nadalje, rezultati pokazuju kako je samoučinkovitost značajno povezana s percepcijom lakoće korištenja.

U Južnoj Africi učenici 10. razreda (10. razred je ekvivalent našem prvom razredu srednje škole) kroz *Scratch* usvajaju osnovne koncepte programiranja, a u 11. razredu (11. razred je ekvivalent našem drugom razredu srednje škole) prelaze na *Javu*, programski jezik više razine. Istraživanje, vođeno modelom prihvaćanja tehnologije (engl. *Technology Acceptance Model*, TAM), pokazalo

je da učenici 10. razreda smatraju *Scratch* korisnim i lakim za korištenje, a učenici 11. razreda smatraju da je jednostavan za korištenje, ali koristan samo u učenju uvodnih koncepata programiranja. *Scratch* pomaže učenicima razumjeti logiku i rješavanje problema, ali ne pomaže im dovoljno u pripremi za korištenje programskog jezika više razine kao što je *Java* (Marimuthu i Govender, 2018).

*Scratch* se može koristiti i *online* i *offline* verziji, pa je vrlo koristan za proces učenja na daljinu (engl. *Distance Learning Process*, DLP). Nakon obuke učitelja za rad u programu *Scratch* provjerena je njihova razina prihvaćanja nove tehnologije. Mjereni pokazatelji bili su *percipirana korisnost*, *percipirana lakoća korištenja*, *namjera korištenja* i *stvarna upotreba tehnologije* te *samo-učinkovitost*. Iz rezultata provedene analize može se vidjeti da namjera korištenja utječe na stvarnu upotrebu tehnologije. Uočena korisnost i percipirana lakoća korištenja zajedno utječu na namjeru stvarne upotrebe, dok uočena lakoća korištenja i samoučinkovitost zajedno imaju značajan učinak na percipiranu korisnost (Gunadi i Sudaryana, 2021).

## **2 Istraživanje**

### **2.1 Metodologija istraživanja**

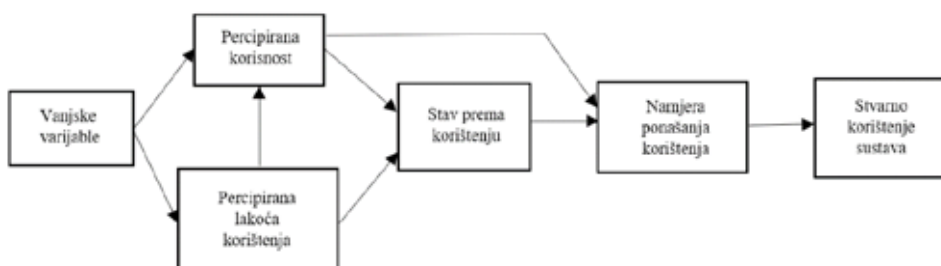
Model prihvaćanja tehnologije (engl. *Technology Acceptance Model*, TAM) teorija je informacijskih sustava koja modelira kako korisnici prihvaćaju i koriste tehnologiju. Stvarna upotreba sustava krajnja je točka u kojoj ljudi koriste tehnologiju, a brojni čimbenici utječu na njihovu odluku o tome kako i kada će je koristiti.

Percipirana korisnost je kognitivna procjena hoće li usvajanje nove tehnologije utjecati na radnu izvedbu pojedinca. Percipirana korisnost utječe na stav pojedinca prema korištenju novih tehnologija jer ljudi imaju tendenciju formirati pozitivne stavove prema novim tehnologijama ako vjeruju da će biti korisna za ono što žele raditi. Percipirana korisnost također izravno utječe na namjeru ponašanja za korištenje sustava. Pojedinci razvijaju namjeru za korištenje tehnologije ako su uvjereni da će ona pozitivno utjecati na radnu uspješnost kao što su povišice ili unapređenja. Motivacija za dobivanjem uvjetnih



nagrada nadilazi osobne osjećaje koje pojedinci imaju prema tehnologiji (Davis i Davis, 1989).

Percipirana lakoća korištenja utječe na stav prema korištenju i na percipiranu korisnost. Kad je tehnologija jednostavna za upotrebu, korisnik ima snažno uvjerenje u svoju sposobnost korištenja sustava, što rezultira entuzijastičnijim stavom prema sustavu. Lakoća korištenja također izravno utječe na izvedbe pojedinca s obzirom na to da će nova tehnologija vjerojatno dovesti do završetka zadatka uz manje napora (Davis i Davis, 1989).



Slika 1. TAM model

Izvor: Davis i Davis, 1989

Prema Davisu i Davisu (1989), percipirana korisnost i percipirana lakoća korištenja imaju značajan utjecaj na stav, što zauzvrat utječe na namjeru stvarnog korištenja. Gledano u smislu TAM okvira, stvarno korištenje *Scratcha* može biti određeno namjerom ponašanja korištenja, a namjera je određena ukupnim stavom učenika prema *Scratchu*. Percepcija korisnosti i lakoće korištenja oblikuju njihov stav (slika 1). U ovom radu percipirana korisnost je prikazana kao stupanj u kojem učenik vjeruje da korištenje programskog jezika *Scratch* poboljšava njegovu sposobnost programiranja. Percipirana lakoća korištenja označava stupanj vjerovanja da je *Scratch* jednostavan za rad bez ulaganja velikog kognitivnog napora. Vanjske varijable, kao što je društveni utjecaj, važni su čimbenici za određivanje stava. Isto tako, percepcija se može promijeniti ovisno o dobi i spolu jer su svi različiti.

Cilj ovog istraživanja je otkrivanje povezanosti četiriju varijabli TAM modela, *percipirane korisnosti*, *percipirane lakoće korištenja*, *stava* i *namjere korištenja programskog jezika Scratch* među učenicima jedne srednje strukovne škole u dobi od 15 do 17 godina. U skladu s navedenim ciljem istraživanja postavljene su sljedeće hipoteze:

- H1 – Motivacijska varijabla *percipirana korisnost* značajan je prediktor varijable *stav prema korištenju*. Varijable su pozitivno povezane.
- H2 – Motivacijska varijabla *percipirana lakoća korištenja* značajan je prediktor varijable *namjere korištenja*. Varijable su pozitivno povezane.
- H3 – Varijabla *stav prema korištenju* značajan je prediktor *namjere korištenja*. Varijable su pozitivno povezane.

Istraživanje je provedeno tijekom redovne nastave Računalstva u školskoj godini 2021./2022. u prirodnom okruženju bez prisustva trećih osoba. Istraživanje je obuhvatilo tri prva razreda (15 – 16 godina) i tri druga razreda (16 – 17 godina) srednje strukovne škole. Provedba istraživanja je trajala dva tjedna i odvijala se u nekoliko etapa: upoznavanje s okruženjem *Scratch*, demonstracija razvoja igrice, samostalan rad učenika, konstruktivno komentiranje vlastitih i tuđih igrica te ispunjavanje izlaznih TAM upitnika.

Kako se nastava tijekom prethodne školske godine 2020./2021. odvijala većinom *online* zbog nepovoljnih uvjeta uzrokovanih virusom Covid, u prvim razredima nije provedena planirana nastavna cjelina programiranju te su učenicima obiju dobnih skupina većinom bili početnici u programiranju. Provedba predmeta Informatika kao izbornog u završnim razredima osnovne škole ima za posljedicu heterogenost učenika po znanju i sposobnostima u srednjoj školi pa se tako u istom razredu mogu nalaziti učenici bez predznanja programiranja, kao i učenici koji su već „iskusni programeri“.

Na početku provedbe nastavne cjeline „Rješavanje problema pomoću računala“ uveden je *Scratch* za upoznavanje učenika s pojmom algoritma. Svi sudionici su bili izloženi istom tretmanu u terminu po dva sata tjedno kao blok-sat. U poučavanju je primijenjen model kognitivnog šegrtovanja. Tijekom prvog sata nastavnica je provela učenike kroz proces registracije i kreiranja korisničkog računa u *Scratchu*, predstavila okruženje *Scratch*, objasnila skripte,

naredbe, događaje i kontrolne strukture te pokazala igricu koju će učenici izrađivati. Učenici su na prvom satu, prateći upute nastavnice, započeli s izradom vlastitog lika i pripadajućeg kostima, kao i pozadine za cijelu igricu. Tijekom sljedećih triju školskih sati učenici su izradili vlastite animacije, u početku uz povremenu pomoć nastavnice, a zatim potpuno samostalno. Kroz izradu jednostavne igrice učenici su se upoznali sa svim trima algoritmima, slijedom, ponavljanjem i uvjetnim grananjem te pojmom varijable. Tema odabrane igrice je bliska učenicima i ciljano prilagođena uzrastu sudionika i interesima učenika kako bi se povećala motivacija. Vizualni izgled naredbi u *Scratchu* omogućava intuitivno usvajanje značenja pojedinih naredbi i slaganje jednostavnih skripti. Zbog vizualnog izvršavanja programa učenici su mogli pratiti izvođenje programa, prepoznati eventualne greške u izvršavanju te ih na primjeren način ispraviti. Nakon upoznavanja s temeljnim konceptima programiranja planiran je prelazak na tekstualni programski jezik *Python* koji je ujedno i zadani programski jezik po godišnjem izvedbenom kurikulumu.

Izlaznim TAM upitnicima izmjerilo se prihvaćanje *Scratcha*, percipirana korisnost, percipirana lakoća korištenja, stav prema korištenju i namjera korištenja. Istraživanje je obuhvatilo i kvalitativne i kvantitativne tehnike prikupljanja podataka. Otvoreni tip pitanja omogućio je razumijevanje prikupljenih kvantitativnih podataka. Kvantitativni podatci korišteni su za analizu povezanosti među četirima varijablama TAM-a (percipirane korisnosti, percipirane lakoće korištenja, stava prema korištenju i namjere korištenja). Za istraživanje je planirana ciljna populacija od 140 sudionika. Tablica 1 prikazuje uzorak sudionika po svakom upitniku koji su u konačnici ispunjeni i vraćeni.

Tablica 1: Uzorak sudionika u istraživanju

Upitnik	broj učenika	dječaci	djevojčice
percipirana korisnost	78	77	1
percipirana lakoća korištenja	72	71	1
stav prema korištenju	76	75	1
namjera korištenja	71	70	1

Za dobivanje vrijednosti svih varijabli predviđenih TAM modelom u ovom istraživanju korišteni su sljedeći instrumenti mjerenja:

- a. Upitnik percipirane korisnosti ispituje procjenu učenika o stupnju korisnosti koje će učenje programiranja u *Scratchu* imati na njegovu ukupnu sposobnost programiranja. Upitnik se sastoji od pet afirmativnih tvrdnji i jednog pitanja otvorenog tipa. Tablica 2 prikazuje opis tvrdnji kao i tip pitanja.

Tablica 2: Upitnik percipirane koristi

pitanje	tvrdnja	tip pitanja
P1	<i>Scratch</i> unapređuje moje logičko razmišljanje.	Likertova skala (1 – 5)
P2	<i>Scratch</i> unapređuje moju sposobnost rješavanja problema.	Likertova skala (1 – 5)
P3	<i>Scratch</i> je koristan za razumijevanje temeljnih koncepata programiranja.	Likertova skala (1 – 5)
P4	Rad u <i>Scratchu</i> povećava moju motivaciju za učenjem programiranja.	Likertova skala (1 – 5)
P6	<i>Scratch</i> razvija moju kreativnost.	Likertova skala (1 – 5)
P7	Ako imaš neki svoj komentar u vezi programiranja u <i>Scratchu</i> , slobodno ga napiši.	otvoreni tip pitanja

- b. Upitnik percipirane lakoće korištenja ispituje procjenu učenika o stupnju lakoće korištenja i snalaženja u programskom okruženju *Scratch*. Upitnik se sastoji od pet afirmativnih tvrdnji. Tablica 3 prikazuje opis tvrdnji, kao i tip pitanja.

Tablica 3: Upitnik percipirane lakoće korištenja

pitanje	tvrdnja	tip pitanja
P1	<i>Scratch</i> je jednostavan za korištenje.	Likertova skala (1 – 5)
P2	Izgled sučelja <i>Scratch</i> je jednostavan i intuitivan.	Likertova skala (1 – 5)
P4	Upute za kreiranje kôda su jednostavne i razumljive.	Likertova skala (1 – 5)
P5	Lako je postati vješt u programiranju u <i>Scratchu</i> .	Likertova skala (1 – 5)
P6	U <i>Scratchu</i> mogu programirati i bez pomoći nastavnika.	Likertova skala (1 – 5)

- c. Upitnik *Stav prema korištenju* mjeri učenikov ukupan stav prema programiranju *Scratch*. Na njegov stav utječu percipirana korisnost i percipirana lakoća korištenja. Tablica 4 prikazuje opis tvrdnji ovog upitnika.

Tablica 4: Upitnik stav prema korištenju

pitanje	tvrdnje	tip pitanja
P1	<i>Scratch</i> ima pozitivan utjecaj na moje programerske vještine.	Likertova skala (1 – 5)
P2	Želio/željela bih više znati o programiranju.	Likertova skala (1 – 5)
P3	Učenje programiranja u <i>Scratchu</i> je dobra ideja.	Likertova skala (1 – 5)

- d. Upitnik namjere korištenja sadrži tri afirmativne tvrdnje (tablica 5). Prema Davisu i Davisu (1989), percipirana korisnost i percipirana lakoća korištenja imaju značajan utjecaj na korisnikov stav, što u konačnici utječe i na njegovu namjeru za stvarnom upotrebom tehnologije.

Tablica 5: Upitnik namjere korištenja

Pitanje	tvrdnje	tip pitanja
P1	Preporučiti ću <i>Scratch</i> drugima.	Likertova skala (1 – 5)
P2	Planiram i u budućnosti programirati u <i>Scratchu</i> .	Likertova skala (1 – 5)
P3	Motiviran/motivirana sam i dalje učiti programiranje.	Likertova skala (1 – 5)

Upitnici sadrže zatvoreni i otvoreni tip pitanja. Zatvorena pitanja koristila su Likertovu skalu ocjena od 1 (*u potpunosti se ne slažem*) do 5 (*u potpunosti se slažem*). Tvrdnje u upitnicima su potvrdno formulirane pa veća ocjena odgovara većem slaganju s tvrdnjom. Otvoreni tip pitanja korišten je za bolje razumijevanje percepcije učenika o *Scratchu*. Ovo istraživanje provedeno je u školskom okruženju tijekom redovnog izvođenja nastave iz predmeta Računalstvo. Učenici prvih i drugih razreda imali su jednake uvjete te nisu imali dodatne domaće zadaće ili bilo kakve dodatne obveze izvan redovne nastave. Istraživanje je naturalističko jer se sudionici nalaze u svom prirodnom okruženju u kojem je istraživač ujedno i nastavnik te samim tim dio tog prirodnog okruženja. Nastava se odvijala po godišnjem izvedbenom kurikulu i od učenika nije bio

potreban svjestan pristanak. S obzirom na to da se radi o naturalističkom istraživanju (Cohen, L., Manion, L. i Morrison, 2011) željelo se eliminirati svjesno mijenjanje ponašanja sudionika koje bi moglo utjecati na valjanost rezultata istraživanja u konačnici. Sudionici su potpuno anonimno odgovarali na upitnike i analizom podataka nije moguće utvrditi njihov identitet. Ispunjavanje upitnika bilo je dobrovoljno i sudionici nisu bili motivirani dobivanjem plusa ili ocjene za aktivnost. Cilj je bio dobiti što iskrenije odgovore na pitanja i komentare. Isto tako, tijekom istraživanja nije bilo ocjenjivanja znanja.

## 2.2 Rezultati i rasprava

Za analizu kvantitativnih podataka korištena je deskriptivna statistička analiza i tehnika korelacijske analize. Vrijednosti deskriptivne statistike svih varijabli, broj ispitanika u istraživanju (N), aritmetička sredina (M), Mod, minimalna i maksimalna izmjerena vrijednost varijable prikazani su u tablici ispod. Učenici su ispunjavali upitnike u *online* obliku nakon tretmana u *Scratchu*. Kako je odgovaranje na upitnike bilo potpuno dobrovoljno, broj učenika (N) je različit za svaki upitnik jer dio učenika nije ispunio sva četiri upitnika.

Tablica 6: Deskriptivna statistika promatranih varijabli

varijabla	N	As	Mod	Min	Max
upitnik percipirane korisnosti	78	3,52	4	1	5
upitnik percipirane lakoće korištenja	72	4,15	4	1	5
upitnik stava prema korištenju	76	3,96	4	1	5
upitnik namjere korištenja	71	3,49	3	1	5

Za mjerenje pouzdanosti upitnika korišten je koeficijent *Cronbach alpha*, čija prihvatljiva vrijednost za empirijska istraživanja iznosi od 0,7 ili više. Upitnici *percipirana korisnost* (Cronbach  $\alpha=0,913$ ) i *percipirana lakoća korištenja* (Cronbach  $\alpha=0,864$ ) spadaju u grupu visoko pouzdanih, dok upitnik *namjera korištenja* spada u grupu granično pouzdanih ( $\alpha=0,685$ ). Za tri pitanja upitnika stav prema korištenju koeficijent pouzdanosti je niži od prihvatljivog ( $\alpha=0,500$ ), što se može pripisati malom broju pitanja u upitniku.

Tablica 7: Rezultati analize pouzdanosti upitnika

varijabla	A
upitnik percipirane korisnosti	0,913
upitnik percipirane lakoće korištenja	0,864
upitnik stava prema korištenju	0,500
upitnik namjere korištenja	0,685

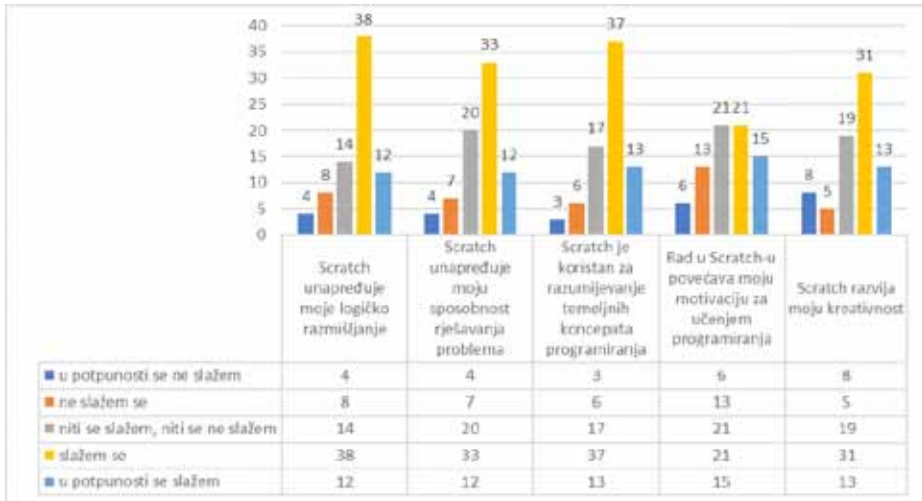
Pearsonov koeficijent korelacije korišten je za mjerenje snage i smjera povezanosti među četirima varijablama modela prihvaćanja tehnologije. Odabrana je Pearsonova korelacija jer se dobiveni kvantitativni podaci statistički značajno ne razlikuju od normalne raspodjele.

Tablica 8 prikazuje deskriptivne podatke za svako pitanje iz upitnika percipirane korisnosti. Deskriptivni podaci su prikazani kao aritmetička sredina svakog pojedinog odgovora i vrijednost odgovora koji se najčešće ponavlja (Mod) te kao minimalna i maksimalna moguća vrijednost odgovora.

Tablica 8: Deskriptivna statistika upitnika percipirane korisnosti

izjava	As	Mod	Min	Max
<i>Scratch</i> unapređuje moje logičko razmišljanje.	3,6	4	1	5
<i>Scratch</i> unapređuje moju sposobnost rješavanja problema.	3,5	4	1	5
<i>Scratch</i> je koristan za razumijevanje temeljnih koncepata programiranja.	3,7	4	1	5
Rad u <i>Scratchu</i> povećava moju motivaciju za učenjem programiranja.	3,3	4	1	5
<i>Scratch</i> razvija moju kreativnost.	3,4	4	1	5

Frekvencije odgovora upitnika percipirana korist prikazane su grafički na slici ispod (slika 2).



Slika 2: Frekvencije odgovora upitnika percipirana korist

Prema frekvencijama odgovora može se zaključiti kako su učenici pozitivno doživjeli korisnost *Scratcha* u pogledu svih tvrdnji osim tvrdnje *Rad u Scratchu povećava moju motivaciju za učenjem programiranja*, gdje je među odgovorima bio gotovo jednak broj nepovoljnih odgovora, ukupno 52 % (27 % ispitanika je bilo neutralno i 25 % se ne slaže ili se u potpunosti ne slaže s tvrdnjom), i povoljnih odgovora, ukupno 48 % (ili slaže se ili se u potpunosti slaže s tvrdnjom). Objašnjenje toga može biti u činjenici da je programiranje danas jedno od najtraženijih, visoko plaćenih zanimanja u svijetu i učenici koji imaju ambiciju učiti „prave“ programske jezike doživjeli su programsko okruženje *Scratch* inferiornim i „dječjim“ u kontekstu učenja programiranja. U prilog toj pretpostavci ide i većina komentara na neobavezno pitanje otvorenog tipa koje je napisalo petero učenika. Komentari su prikazani u obliku kako su ih učenici napisali.

- *scratch je jako dosadan i jednostavan program.*
- *Scratch je zabavan program za programiranje ali nije bas puno ucinkovit.*
- *dobar za pocetnike*
- *Djecja aplikacija za programiranje*
- *vrh*

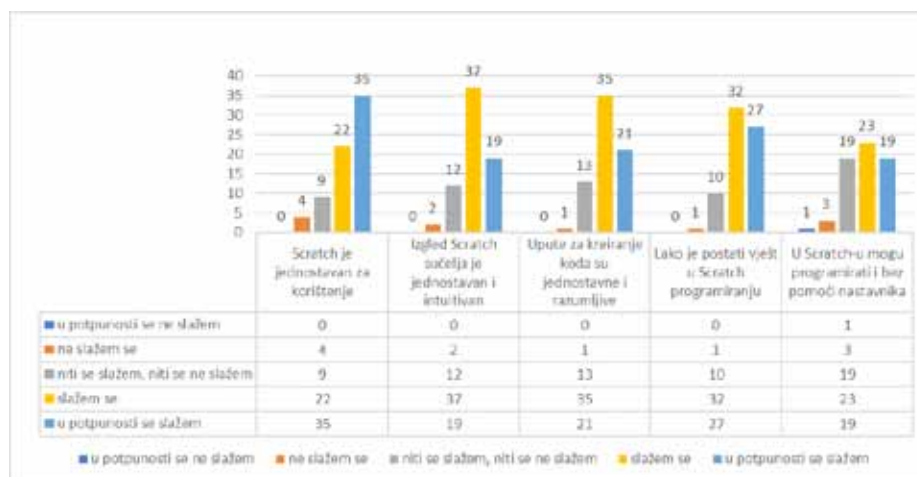


Gotovo jednak broj ispitanika se složio i u potpunosti se složio s tvrdnjama *Scratch unapređuje moje logičko razmišljanje*, *Scratch je koristan za razumijevanje temeljnih koncepata programiranja* i *Scratch poboljšava moje sposobnosti rješavanja problema*, pa se može izvući zaključak da su učenici uvidjeli korisnost učenja programiranja u *Scratchu*. Prema Davisu i Davisu (1989), percipirana korisnost ima značajan utjecaj na oblikovanje stava prema korištenju *Scratcha*, što zauzvrat utječe i na njihovu namjeru stvarnog korištenja.

Tablica 9 prikazuje deskriptivne podatke za svako pitanje iz upitnika percipirane lakoće korištenja. Najčešći odgovor na pitanja ovog upitnika je odgovor *slažem se*.

Tablica 9: Deskriptivna statistika upitnika percipirane lakoće korištenja

Izjava	As	Mod	Min	Max
<i>Scratch je jednostavan za korištenje.</i>	4,2	5	2	5
Izgled sučelja <i>Scratch</i> je jednostavan i intuitivan.	4,0	4	2	5
Upute za kreiranje koda su jednostavne i razumljive.	4,0	4	2	5
Lako je postati vješt u programiranju <i>Scratch</i> .	4,2	4	2	5
U <i>Scratchu</i> mogu programirati i bez pomoći nastavnika.	3,8	4	1	5



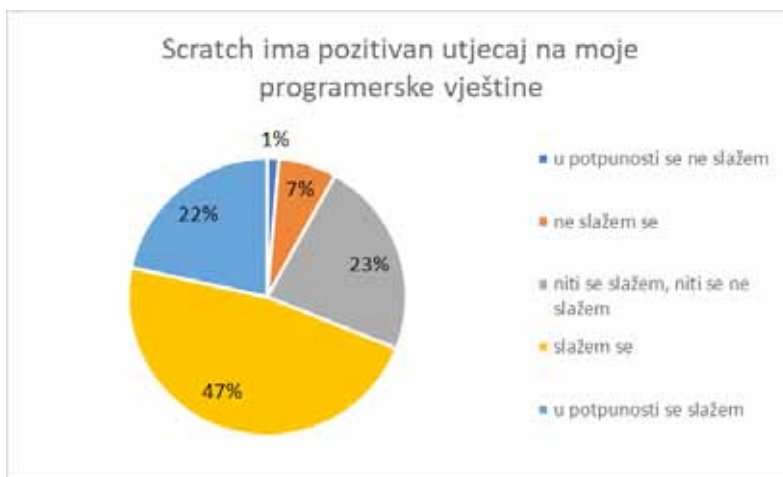
Slika 3: Frekvencije odgovora upitnika percipirana lakoća korištenja

Učenici imaju uglavnom pozitivne odgovore (u potpunosti se slažu ili slažu) u pogledu tvrdnji jednostavnosti korištenja, što ukazuje na činjenicu da je većini učenika *Scratch* jednostavan za korištenje (slika 3). Jedina razlika u ovom pozitivnom nizu odgovora je posljednja tvrdnja: *U Scratchu mogu programirati i bez pomoći nastavnika*, gdje je 36 % učenika dalo nepovoljan odgovor (19 učenika je ostalo neutralno, 3 učenika se nisu složila i 1 učenik se u potpunosti nije složio). Budući da je većina učenika prvi put bila izložena programiranju u *Scratchu*, logično je da se osjećaju sigurnijima uz nastavnikovo vođenje i ovi odgovori nisu neočekivani.

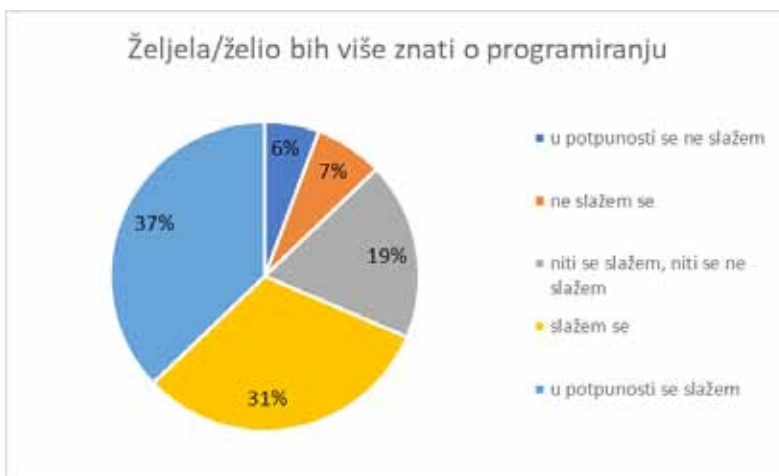
Tablica 10 prikazuje deskriptivne podatke za upitnik *Stav prema korištenju*. Većina učenika ima pozitivan stav prema korištenju *Scratcha*. Najčešći odgovori na sva tri pitanja ovog upitnika su ili *slažem se* ili *u potpunosti se slažem*.

Tablica 10: Deskriptivna statistika upitnika stav prema korištenju

izjava	As	Mod	Min	Max
<i>Scratch</i> ima pozitivan utjecaj na moje programerske vještine.	3,8	4	1	5
Željela/želio bih više znati o programiranju.	3,8	5	1	5
Učenje programiranja u <i>Scratchu</i> je dobra ideja.	3,7	4	1	5



Slika 4: Utjecaj Scratcha na razvoj računalnog razmišljanja



Slika 5: Motivacija za daljnje učenje programiranja nakon tretmana u Scratchu



Slika 6: Dojam o programiranju u Scratchu

Ukupni pozitivni odgovori (*u potpunosti se slažem* i *slažem se*) na svako pojedino pitanje iz ovog upitnika se nalaze u rasponu od 64 % do 69 % (slika 4), (slika 5), (slika 6). Nepovoljni odgovori (*niti se slažem niti se ne slažem*, *ne slažem se* i *u potpunosti se ne slažem*) nikad nisu bili u većini, s manje od 40 % ukupno. Učenici su većinom imali pozitivne odgovore u upitnicima percipirane

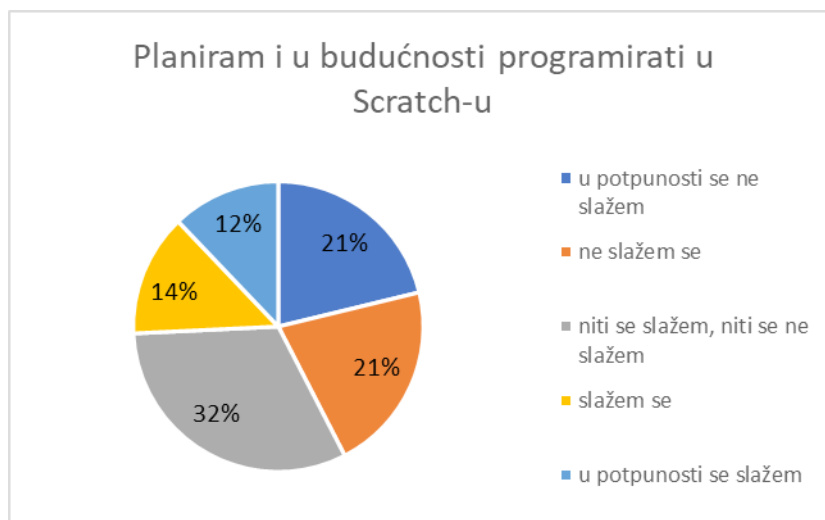
korisnosti i percipirane lakoće korištenja, dvije motivacijske varijable koje imaju značajan utjecaj na stav, pa je ovakav rezultat i očekivan.

Tablica 11 prikazuje deskriptivne statističke podatke za namjeru prema korištenju *Scratcha*. Prema rezultatima, može se primijetiti da je najčešći odgovor na prva dva pitanja niti se slažem niti se ne slažem, što znači da su učenici neutralni u svojoj namjeri korištenja *Scratcha*.

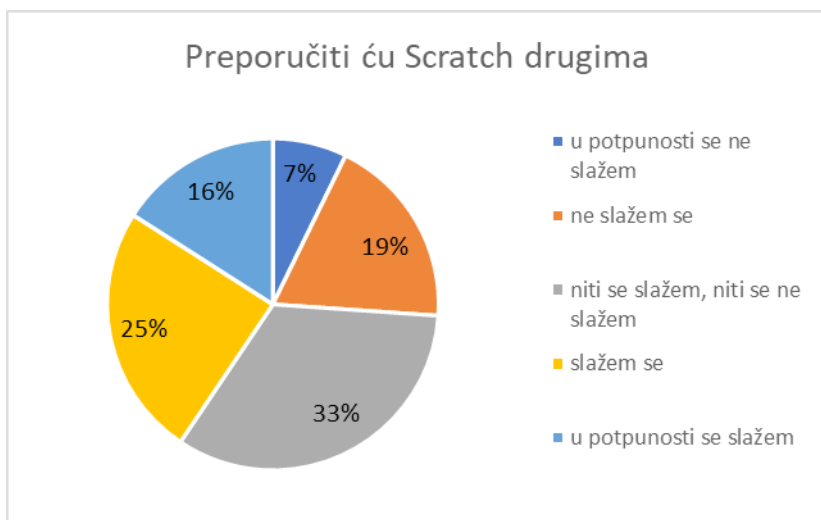
Tablica 11: Deskriptivna statistika upitnika namjere korištenja

izjava	As	Mod	Min	Max
Preporučiti ću <i>Scratch</i> i drugima.	4,0	3	1	5
Planiram i u budućnosti programirati u <i>Scratchu</i> .	2,7	3	1	5
Motiviran/Motivirana sam i dalje učiti programiranje.	3,9	5	1	5

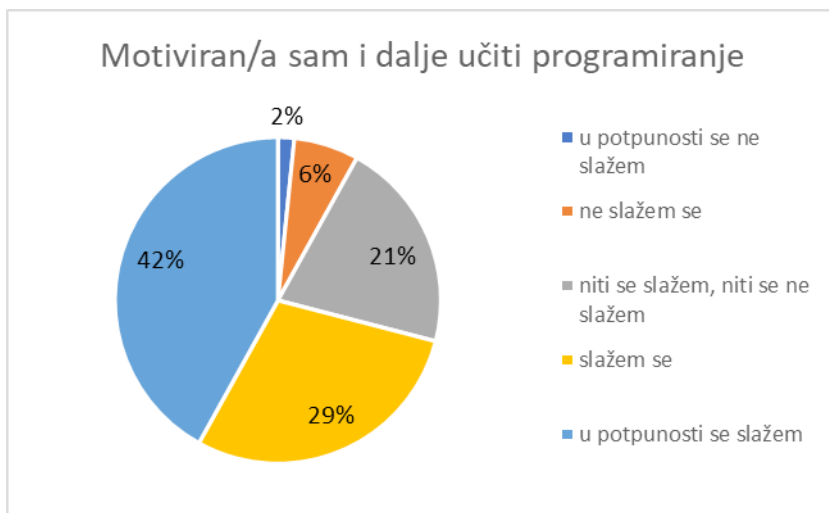
Na slikama ispod grafički su prikazani postotci odgovora na svako pojedino pitanje iz ovog upitnika.



Slika 7: Mišljenje o nastavku programiranja u Scratchu



Slika 8: Mišljenje o preporuci Scratcha drugima



Slika 9: Motivacija za nastavak učenja programiranja

Složilo se ili se u potpunosti složilo s izjavom da su motivirani i dalje učiti programiranje 71 % ispitanika (slika 9). Međutim, bilo je mnogo nepovoljnih odgovora na izjave *Planiram i u budućnosti programirati u Scratchu* (samo 26 % se složilo ili u potpunosti složilo) i *Preporučiti ću Scratch drugima* (samo

41 % se složilo ili u potpunosti složilo). Za te dvije tvrdnje većina ispitanika je ostala ili neutralna ili se nije složila. Neutralna su 32 % ispitanika, dok se 42 % nije složilo ili u potpunosti složilo s tvrdnjom *Planiram u budućnosti programirati u Scratchu* (slika 7). Neutralno je 33 % ispitanika, dok se 26 % ispitanika nije složilo s tvrdnjom *Preporučit ću Scratch drugima* (slika 8). Dakle, to sugerira da učenici početnih razreda srednje škole ipak percipiraju *Scratch* kao „dječji alat za programiranje“ te su svjesni njegovih ograničenja i sukladno tomu su shvatili da ga ne namjeravaju koristiti u budućnosti.

Pearsonov koeficijent korelacije je korišten za mjerenje snage i smjera za sve četiri TAM varijable. Pearsonov koeficijent može varirati od +1 do -1. Dvije varijable su u pozitivnoj korelaciji ako između njih postoji izravna, jednosmjerna veza.

**Tablica 12:** Korelacijska analiza upitnika

		Korelacije			
		Stav prema korištenju	Percipirana lakoća korištenja	Percipirana korisnost	Namjera korištenja
Stav prema korištenju	Pearson-ov koeficijent korelacije	1	.102	.302*	.154
	Statistička značajnost		.405	.012	.207
	Broj ispitanika (N)	69	69	69	69
Percipirana lakoća korištenja	Pearson-ov koeficijent korelacije	.102	1	.038	.279*
	Statistička značajnost	.405		.759	.020
	Broj ispitanika (N)	69	69	69	69
Percipirana korisnost	Pearson-ov koeficijent korelacije	.302*	.038	1	.139
	Statistička značajnost	.012	.759		.253
	Broj ispitanika (N)	69	69	69	69
Namjera korištenja	Pearson-ov koeficijent korelacije	.154	.279*	.139	1
	Statistička značajnost	.207	.020	.253	
	Broj ispitanika (N)	69	69	69	69

\*. Correlation is significant at the 0.05 level (2-tailed).

Dobiveni rezultati (tablica 12) pokazuju postojanje statistički značajne pozitivne povezanost (.302\*) između uvjerenja učenika da je *Scratch* koristan i njihova stava prema korištenju. Ova statistički značajna povezanost pokazuje da je percipirana korisnost *Scratcha* značajno utjecala na njihov stav o *Scratchu*.

Uvjerenje ispitanika o lakoći korištenja *Scratcha* također je značajno koreliralo (.279\*) s namjerom ponašanja učenika da ga koriste. To potvrđuje postojanje linearne povezanosti između lakoće korištenja *Scratcha* i stvarne namjere korištenja *Scratcha*. Dobivene korelacije su u skladu s TAM modelom i dosadašnjim istraživanjima. Percipirana lakoća korištenja utječe na namjeru ponašanja korištenja. Percipirana korisnost utječe na stav prema korištenju i time su potvrđene prve dvije postavljene hipoteze ovog istraživanja.

Međutim, varijabla *stav* nije značajno korelirala (.154) s namjerom ponašanja korištenja, što ukazuje na to da stav učenika o *Scratchu* nije značajno utjecao na njihovu namjeru korištenja. Time je treća hipoteza samo djelomično potvrđena jer je povezanost između varijabli ipak pozitivna.

### 3 Zaključak

Cilj provedenog istraživanja bio je otkriti kako srednjoškolci u dobi od 15 do 17 godina prihvaćaju programiranje u *Scratchu* istražujući povezanost četiriju varijabli TAM modela: *percipiranu korisnost*, *percipiranu lakoću korištenja*, *stav prema korištenju* i *namjeru korištenja*.

Rezultati istraživanja pokazuju da je percipirana lakoća korištenja značajan čimbenik u učenikovo namjeri stvarnog korištenje *Scratcha*. Analiza rezultata također je pokazala da je percipirana korisnost značajno utjecala na učenički stav prema korištenju *Scratcha*.

Kad se odgovori prikupljeni iz anketnih pitanja razmatraju zajedno, može se zaključiti da učenici početnih razreda srednje škole doživljavaju *Scratch* jednostavnim za korištenje i korisnim, ali ipak nedovoljno ozbiljnim alatom za učenje programiranja. Takvo mišljenje učenika o *Scratcha* može se pripisati činjenici da su ti učenici bili izloženi drugim programskim jezicima tijekom osnovnoškolskog obrazovanja i mnogi su došli do uvjerenja da *Scratch* ne pruža

potrebno znanje, sintaktičko i računalno, potrebno za „pravo“ programiranje u proceduralnim jezicima.

Dakle, ukupni podaci prikupljeni ovim istraživanjem sugeriraju pozitivan utjecaj *Scratcha* na motivaciju za daljnjim učenjem programiranja u nekom „ozbiljnijem“ programskom jeziku. To je ujedno i bio cilj uvođenja vizualno-blokovskog jezika na početku nastavne cjeline programiranja kao posredovano-g transфера iz jednog konteksta programiranja u drugi.

Ograničenje ovog istraživanja je kratko vrijeme provođenja tretmana. Cijelo istraživanje je trajalo dva tjedna, odnosno četiri školska sata. Dodatno ograničenje je sudjelovanje manjeg broja učenika pa bi bilo zanimljivo provesti istraživanje na većem broju učenika i razreda u različitim srednjim školama.

Kako je nastavnica Računalstva ujedno i provodila istraživanje, imala je i ulogu sudjelujućeg opažača. Najvažniji zaključak je da *Scratch* omogućuje uvođenje temeljnih koncepata programiranja u puno manje vremena za razliku od tekstualnih programskih jezika. U ovom istraživanju učenici su se upoznali s algoritmom slijeda, ponavljanja i grananja tijekom četiriju školskih sati. S algoritmom slijeda i ponavljanja već tijekom prvog sata programiranja kroz simulaciju pokretanja i upravljanja likom. Pojam varijable, kao jedan od najtežih koncepata programiranja, uveden je već na četvrtom školskom satu kroz skupljanje bodova u igrici. Svi navedeni koncepti su učenicima predstavljeni kroz kontekst igrice pa je njihova uloga vrlo konkretna i jasna i učenik razumije program. Rezultat izvršavanja naredbi vidljiv je kroz kretanje likova i učenici su planirali rješenja programa, promatrali su izvršavanje programa, samostalno uočavali i otklanjali pogreške. Okruženje *Scratch* omogućuje usmjeravanje pažnje na rješavanje problema bez suočavanja s problemom krute sintakse. Grupe naredbi su kodirane bojom i oblikom pa je svaka naredba intuitivno jasna čak i bez dodatnog pojašnjavanja. Na taj način su i učenici sa slabije razvijenim apstraktnim razmišljanjem aktivno sudjelovali u nastavi, dok su samostalniji učenici imali više vremena istraživati dodatne naredbe kojima mogu unaprijediti svoju igricu. *Scratch* potiče i učeničku suradnju, komunikaciju, razmjenjivanje i uspoređivanje ideja te međusobno pomaganje u rješavanju problema pri pisanju programa.



Kad se sagledaju razmišljanja srednjoškolaca obuhvaćenih ovim istraživanjem, može se jasno reći da učenici vole programiranje i žele učiti programirati. Isto tako, većina učenika je doživjela *Scratch* jednostavnim za korištenje i, još važnije, korisnim za razvoj sposobnosti rješavanja problema i logike. Ovakvi nalazi potiču potrebu za razvojem multimedijских aktivnosti i okruženja koji bi, posebno početnicima u učenju programiranja, olakšali prijelaz u svijet „pravog“ programiranja u tekstualnom programskom jeziku. S druge strane, samo osiguravanje takvog okruženja nije dovoljno za učinkovito poučavanje. Zadatke i izazove za učenike treba oblikovati prema njihovoj dobi i interesima, ali i kontekstu vremena u kojem živimo, pri čemu su također potrebni istraživanja u edukaciji, dokumentirana najbolja praksa, standardi kurikula, profesionalni razvoj i podrška nastavnicima.

## **Literatura**

Armoni, M., Meerbaum-Salant, O. i Ben-Ari, M. (2015). From scratch to “Real” programming. *ACM Transactions on Computing Education*, 14(4). <https://doi.org/10.1145/2677087>

Arpaci, I., Durdu, P. O. Mutlu, A. (2019). The Role of Self-Efficacy and Perceived Enjoyment in Predicting Computer Engineering Students’ Continuous Use Intention of Scratch. *International Journal of E-Adoption*, 11(2). <https://doi.org/10.4018/ijea.2019070101>

Ben-Ari, M. (1998a). Constructivism in Computer Science Education. *Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education*, 257–261. <https://doi.org/10.1145/273133.274308>

Ben-Ari, M. (1998b). Constructivism in Computer Science Education. *SIGCSE Bull.*, 30(1), 257–261. <https://doi.org/10.1145/274790.274308>

Bonar, J. i Soloway, E. (1983). Uncovering Principles of Novice Programming. *Proceedings of the 10th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, 10–13. <https://doi.org/10.1145/567067.567069>

Bubica, N., Mladenović, M. i Boljat, I. (2013). *Programiranje kao alat za razvoj apstraktnog mišljenja*.

Cohen, L., Manion, L. i Morrison, K. (2011). *Research Methods in Education (7th ed.)*. Retrieved from <https://doi.org/10.4324/9780203720967>

Davis, F. i Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13, 319-. <https://doi.org/10.2307/249008>

Gomes, A. J. i Mendes, A. J. N. (2007). Problem Solving in Programming. *PPIG*.

Gunadi, G. i Sudaryana, I. K. (2021). ANALISA TINGKAT PENERIMAAN APLIKASI SCRATCH MENGGUNAKAN TECHNOLOGY ACCEPTANCE MODEL (TAM). *Infotech: Journal of Technology Information*, 7(1). <https://doi.org/10.37365/jti.v7i1.101>

Jiang, B. i Li, Z. (2021). Effect of Scratch on computational thinking skills of Chinese primary school students. *Journal of Computers in Education*, 8(4). <https://doi.org/10.1007/s40692-021-00190-z>

Lai, A. F. i Yang, S. M. (2011). The learning effect of visualized programming learning on 6th graders’ problem solving and logical reasoning abilities. *2011 International*

*Conference on Electrical and Control Engineering, ICECE 2011 - Proceedings*. <https://doi.org/10.1109/ICECENG.2011.6056908>

Maloney, J., Resnick, M., Rusk, N., Silverman, B. i Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4). <https://doi.org/10.1145/1868358.1868363>

Marimuthu, M. i Govender, P. (2018). Perceptions of Scratch Programming among Secondary School Students in KwaZulu-Natal, South Africa. *The African Journal of Information and Communication*, (21). <https://doi.org/10.23962/10539/26112>

Marques, F. O. i Marques, M. T. (2012). No problem ? No research, little learning ...Big problem! *Journal of Systematics, Cybernetics i Informatics*, 10(3).

Milne, I., i Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information Technologies*, 7, 55–66. <https://doi.org/10.1023/A:1015362608943>

Monika Mladenović. (2019). *Poučavanje početnog programiranja oblikovanjem računalnih igara*.

Permatasari, L., Yuana, R. A. i Maryono, D. (2020). Implementation of Scratch Application to Improve Learning Outcomes and Student Motivation on Basic Programming Subjects. *Journal of Informatics and Vocational Education*, 3(1). <https://doi.org/10.20961/joive.v2i2.35760>

Resnick, M., Maloney, J., Hernández, A. M., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. (2009). Scratch: Programming for Everyone. *Communications of the ACM*, 52.

Tan, P. H., Yee, T. i Ling, siew-woei. (2009). *Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception*. 42–46. <https://doi.org/10.1109/ICCTD.2009.188>

Theodorou, C. i Kordaki, M. (2010). Super Mario : a Collaborative Game for the Learning of Variables in Programming. *Building*, 2(4).

Topalli, D. i Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers and Education*, 120. <https://doi.org/10.1016/j.compedu.2018.01.011>

Wilson, A. i Moffat, D. (2012). Evaluating Scratch to introduce younger schoolchildren to. *Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest Group-PPIG2010, September 19-22, 2010*.