

Toni Mrđen¹

mr. sc. Ivan Livaja, v. pred.¹

Jerko Acalin, dipl. ing., pred.¹

PROGRAMSKI JEZIK SCRATCH - PRIMJENA U EDUKACIJI

Stručni rad / Professional paper

UDK 004.43

Računalna tehnologija nezaustavljivom brzinom prodire u svaki djelić ljudskog života i rada. Djeca nove tehnologije primaju na drugačiji način od odraslih, jer odrastaju s novim izazovima. Uloga učitelja je usmjeriti učenike kako kreativno koristiti informacijsku tehnologiju, te njome podignuti razinu obrazovanja u cjelini. Informatika je idealno područje kroz koje se kod djece mogu razvijati vještine koje su nužne za cjeloživotno učenje, poput logičkog, stvaralačkog i apstraktnog mišljenja, kreativnosti, učenja metodom pokušaja i pogrešaka, stalnim dopunjavanjem i ispravljanjem vlastitog rješenja te korištenjem različitih izvora znanja. Korištenje osnovnih alata za rad na računalu je svakako korisno, kao što je korisno i poznavati osnove programiranja. I dok danas djeca, sutra odrasli, možda neće kroz život morati programirati, dobro će im doći praktično znanje o načinu na koji funkcioniraju tehnologije koje koriste i okvirno shvaćanje toga što s njima mogu postići. Jedan od alata za početno učenje programiranja je Scratch, programski jezik i razvojno okruženje za stvaranje priča, igara, animacija, glazbe i drugih interaktivnih sadržaja na računalu, posebno osmišljen kako bi bio pristupačan i zanimljiv djeci. U ovom radu bit će objašnjen način programiranja aplikacija u Scratchu, te će na jednostavnim primjerima biti prikazana struktura programa sastavljena od grafičkih blokova koji predstavljaju naredbe.

Ključne riječi: Scratch, programiranje, apstraktno mišljenje.

1. Uvod

Pojavom osobnog računala u sedamdesetima, došlo je do veće popularizacije programiranja. Rani počeci programiranja bili su pretežito vezani uz matematiku i njegova se edukacija obično vršila samo na sveučilištima.

U današnje vrijeme, djeca odrastaju uz tehnologiju te je ona sveprisutna u njihovoj okolini. O računalima i informacijskoj tehnologiji uče od malih nogu jer su okruženi njome i shodno tome, povećana je edukacija programiranja od učenja informatike u osnovnim i srednjim školama do izvanškolskih radionica i edukacija. Savladati programiranje je težak i zahtjevan poduhvat. Mnogi početnici koji uopće nemaju iskustva s programiranjem misle da

¹ Veleučilište u Šibeniku, Šibenik

je programiranje samo monotono pisanje koda satima pred računalom, ali programiranje je puno više od toga. Programiranje je rad na rješavanju problema, otklanjanje grešaka u kodu, razvijanje i mijenjanje logičkog razmišljanja. Ono je apstraktno te je samim time djeci teže za naučiti, ali je na drugu stranu odlično za razvijanje apstraktnog mišljenja.

Pomoć učenicima početnicima u osnovnim školama te izvanškolskim radionicama i edukacijama pruža se kroz vizualne programske jezike, koji otklanjaju probleme ne shvatljive sintakse kroz vizualne reprezentacije. Za odabir prvog programskog jezika, potrebno je razmotriti određene kriterije. Jezici koji nisu pogodni za učenje programiranja, obično imaju kompliciranu sintaksu i zbunjujuću semantiku. Simboli i nepotrebne kratice koje u jeziku mogu zbuniti učenike u samim početnicima ostavlja loš dojam o jeziku, a sukladno i smanjenu motivaciju učenja. Problem teške sintakse dovodi do velike praznine između učenja sintakse i učenja programiranja kroz algoritme.

2. Programiranje

Rani počeci programiranja bili su pretežito vezani uz matematiku i njegova se edukacija obično vršila samo na sveučilištima.

Za razumijevanje tehnologije, potrebno je razumjeti jezik računala, odnosno ovladati principom rada računala. Najbolji način za to je učenje programiranja od najranije dobi, ali na način koji je djeci „prirodan“. Činjenicu da djeca mnogo svog vremena provode za računalom potrebno je iskoristiti kako bi to vrijeme mogli provesti učeći (Bubica, N., Mladenović, M., Boljat, I., 2014)

Programiranje nije samo pisanje računalnih programa. Programiranje je rješavanje problema, otklanjanje grešaka, razvijanje logičkog razmišljanja i računalnog razmišljanja, a to podrazumijeva razvoj strategija za rješavanje problema koji se mogu odnositi i na neprogramerska područja. Zbog toga se može reći da programiranje mijenja način razmišljanja. Kada uzmemo u obzir da živimo u digitalnom dobu i da je djeci tehnologija „prirodno“ okruženje, jasno je da bi takve promjene trebali uzeti u obzir u školovanju, s obzirom na sveprisutnost informatike u svakodnevnom životu i s pretpostavkom da će ta prisutnost postajati sve izraženija (Bubica, N., Mladenović, M., Boljat, I., 2014).

2.1. Programski jezici u edukaciji

Glavni cilj svakog programskog jezika koji je namijenjen učenju programiranja mora biti usvajanje osnovnih znanja i koncepta programiranja koji se mogu primijeniti u ostalim programskim jezicima, a ne samo u jeziku namijenjenom učenju programiranja. Najčešće se u školama i na radionicama (u RH) za učenje programiranja koriste programski jezici: LOGO, Basic, Pascal, Scratch, C++ te Python.

Jedni od najprikladnijih jezika za učenje programiranja su Python i Eiffel. Prema ispunjenosti kriterija Java je zauzela treće mjesto koja je prvenstveno namijenjena za poduzeća i komercijalni software što je ujedno i glavna mana. Neki od kriterija za odabir programskog jezika za učenje su: pogodan za nastavu, nudi opći okvir, promovira novi pristup učenju programiranja, promovira pisanje ispravnih programa, dopušta rješavanje problema koji se

moгу riješiti u malim dijelovima koda (modularnost), nudi bespriječna i jednostavna razvojna okruženja, podržan na različitim okruženjima, ima korisničku podršku kroz zajednice, otvorenog koda, ima prateći adekvatni nastavni materijal, ne koristi se samo u obrazovanju.

3. 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, a omogućuje lako kreiranje interaktivnih priča, igrice, animacija i projekata te njihovo dijeljenje s drugim korisnicima preko web-a. (Mujačić, 2015).

Grupa Lifelong Kindergarten surađivala je s tvrtkom Lego na izradi robota Lego Mindstorms koji se, također, koriste za poučavanje programiranja. Uočilo se da djeci u radu s kockicama odmah počinju navirati ideje, mašta i kreativnost pa su napravili vizualni programski jezik koji podsjeća na slaganje kockica. Naredbe su napravljene u obliku slagalica, tako da je vizualno jasno koje se naredbe mogu složiti. Grupirane su tematski, a razlikuju se po obliku i bojama (Bubica i sur., 2014).

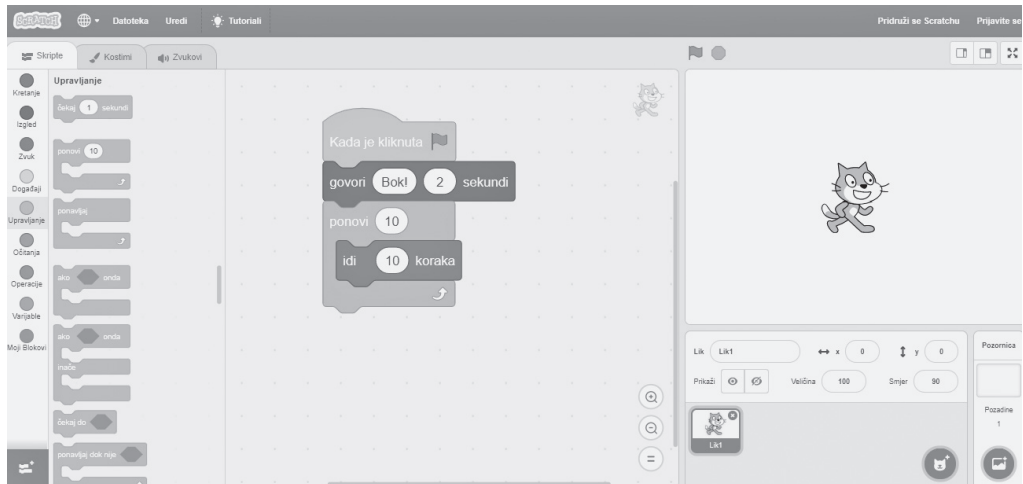
Iako je nastao na idejama programa LOGO, programiranje u Scratchu razlikuje se od programiranja u drugim vizualnim programskim okolinama, jer se u njemu rabe naredbene strukture u obliku grafičkih programskih blokova, pa se na taj način eliminira mogućnost sintakasnih pogrešaka. Program omogućuje programiranje mišem povlačenjem i uklapanjem blokova koji se mogu spojiti samo ako to odgovara u određenom sintaksnom smislu i tako omogućuje učenicima da se fokusiraju na probleme koje oni žele riješiti.

3.1. Osnovni elementi Scratch projekta

Scratch razvojno okruženje nalazi se na jednom prozoru i elementi su fiksni s više pojedinih dijelova. Može se otvoriti na bilo kojem internet pregledniku uz dodatak Adobe Flash. Na desnoj strani postoji mjesto gdje se izvodi program, ispod toga se nalazi popis objekata u programu, na lijevoj strani se nalazi dio s pomoćnim blokovima i opcijama, a u sredini je prostor za slaganje skripti. Postoji mogućnost povećavanja mjesta gdje se program izvodi na cijelom ekranu. Scratch projekt čine objekti koji se zovu likovi (*sprites*). Njihov se izgled može mijenjati dodavanjem različitih kostima (*costumes*). Lik može izgledati poput osobe, vlaka, leptira ili bilo čega drugoga.

Za kostim je moguće koristiti bilo koju sliku; izraditi crtež u u Paint Editor-u, može se učitati slika s računala ili metodom uhvati-povuci-pusti preuzeti sliku s internetske stranice. Liku se mogu dati različite upute: da se kreće, razgovara, reproducira glazbu ili ostvaruje interakciju s drugim likovima. Za izdavanje naredbi potrebno je složiti grafičke blokove u cjeline, nazvane skripte.

Slika 1: Razvojno sučelje Scratcha



Izvor:screenshotScratch editora (<https://scratch.mit.edu/projects/editor>)

Scratch blokovi organizirani su u devet kategorija različitih boja: Kretanje, Izgled, Zvuk, Događaji, Upravljanje, Očitavanja, Operacije, Varijable i Moji Blokovi. Svaka vrsta blokova ima jedinstven izgled kojim pokazuje koji blok ide ispod, iznad ili unutar na način da se blokovi spajaju kao slagalice. Svojim izgledom blokovi sugeriraju koja su ograničenja i mogućnosti te na taj način onemogućavaju javljanje pogrešaka. Iako nema dojava pogrešaka, sam program može funkcionirati na „krivi način“, ali eksperimentiranjem i iskustvom može se doći do željenih rezultata. Klikom na skriptu, Scratch prolazi kroz blokove od vrha do dna. Ako korisnik samo želi provjeriti kako djeluje određeni blok, dovoljno je kliknuti na njega i lik na ekranu će naredbu odmah izvesti. Skripte koje se grade uvijek se mogu izvesti te pojedini blokovi neće smetati ako se odvoje van skripte.

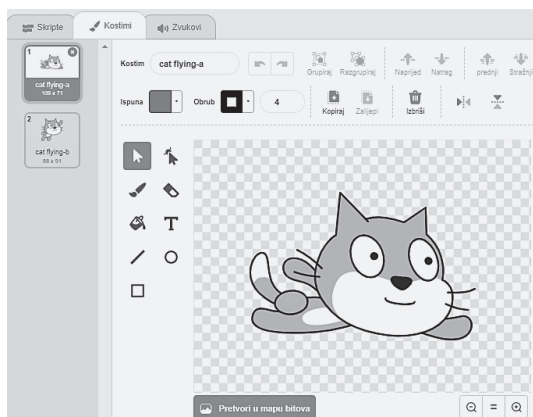
Likovi kojima se daju naredbe nalaze se na pozornici. Na pozornici se likovi pomiču i ostvaruju interakciju, a sam izgled pozornice može se mijenjati. Pozornica je široka 480, a dugačka 360 jedinica. Podijeljena je na x–y os.

Središte Pozornice ima x-koordinatu 0 i y-koordinatu 0. Kako bi se otkrile x–y koordinate na Pozornici, potrebno je pomicati miš (kursor) i pratiti x–y prikaz koji se nalazi ispod Pozornice.

3.2. Kostimi i pozadine

Scratch prepoznaje brojne formate slika: JPG, BMP, PNG, GIF (uključujući i animirani GIF). Svaki kostim ima broj koji je prikazan s njegove lijeve strane. Redoslijed kostima moguće je promijeniti povlačenjem sličica, a njihov će se broj automatski ažurirati ovisno o položaju.

Slika 2: Kartica „kostimi“



Izvor: screenshotScratch editora (<https://scratch.mit.edu/projects/editor>)

Svaki lik ima jedan ili više kostima. S više sličica kostima moguće je napraviti animaciju lika te što više kostima lik ima to je animacija bolja. Zatamnjeni kostim je onaj koji se trenutno koristi u radu.

Pod karticom „kostimi“ postoji i paint editor kojim se može promjeniti lika, ali i nacrtati novog. Postoje gumbi za *Označi*, *Preoblikuj*, *Kist*, *Gumica*, *Ispuna*, *Tekst*, *Linija*, *Kružnica* te *Pravokutnik*. Opcija uređenja i crtanja vlastitih likova obogaćuje Scratch jer kod djece budi kreativnost da naprave svoj vlastiti i unikatni lik.

Kako bi likovi došli do izražaja te cijeli projekt ljepše izgledao, moguće je koristiti već ponuđene, raznolike pozadine raspoređene po kategorijama ili učitati/nacrtati/uslikati svoju te na nju dodavati razne likove.

4. Programske strukture u Scratchu

Programska struktura je ustrojstvo programa, tj. način i redoslijed rješavanja pojedinih manjih radnji da bi se došlo do konačnog rješenja zadatka. Nekoliko je osnovnih programskih struktura koje se pojavljuju u mnogim računalnim programima pa tako i u Scratchu. To su: pravocrtna programska struktura (slijed/sekvencija), struktura grananja (odabir/selekcija) struktura petlje (ponavljanje/iteracija).

4.1. Pravocrtna programska struktura (slijed/sekvencija)

Slijed ili sekvencija (eng. Sequention) je najjednostavnija logička struktura u kojoj se naredbe odvijaju jedna iza druge, onim redom kako su napisane, a formiraju se u obliku bloka naredbi (Lipljin, 2004, str. 111).

Slika 3. Pravocrtna programska struktura



Izvor: screenshotScratch editora (<https://scratch.mit.edu/projects/editor>)

4.2. Struktura grananja (odabir/selekcija)

Odabir ili selekcija (eng. Selection) je mjesto u programu na kojem se ispituje neka vrijednost ili logički izraz koji može biti jednostavan ili složen. Ovisno o rezultatu ispitivanja, dolazi do grananja ili do jednostrane selekcije (Lipljin, 2004, str. 111).

Grananje je programska struktura koja omogućuje različit tijek programa, ovisan o rezultatu postavljenog uvjeta. To je važna struktura bez koje bi mogućnost rješavanja zadataka računalom bila vrlo ograničena. Rezultat postavljenog uvjeta mora biti jedno od dva stanja: true/false, da/ne, istina/laž, 1/0. Vrijednost rezultata uvjeta bit će 1 ako je uvjet zadovoljen i 0 ako uvjet nije zadovoljen.

Naredba If radi tako da se ispita uvjet i u skladu sa istinitom ili lažnom vrijednošću uvjetnog izraza izvršavaju naredbe. Ukoliko je uvjet istinit, naredba ili blok naredbi koje slijede se izvršavaju, a ako je neistinit naredbe se preskaču.

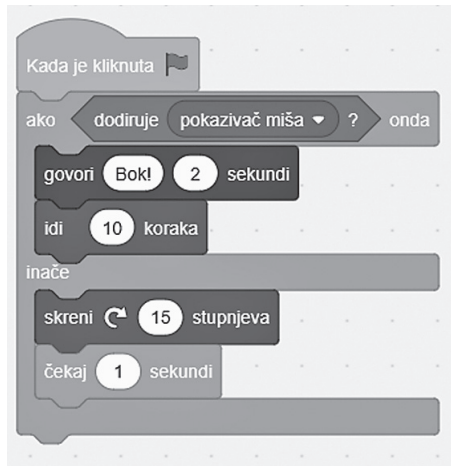
Slika 4: If naredba



Izvor: screenshotScratch editora (<https://scratch.mit.edu/projects/editor>)

Naredba If/else omogućava da se ovisno o ispunjenju postavljenog uvjeta izvodi jedan od dva neovisna bloka naredbi. Ako je vrijednost uvjeta istina, izvodi se prvi blok naredbi. Ako je vrijednost uvjeta neistina, preskače se prvi blok naredbi i izvodi se drugi blok (iza naredbe else).

Slika 5: If/else naredba



Izvor: screenshotScratch editora (<https://scratch.mit.edu/projects/editor>)

4.3. Struktura petlje (ponavljanje/iteracija)

Ako se neki dijelovi programa ponavljaju određen broj puta, govori se o logičkoj strukturi ponavljanja ili iteracije (eng. Loop) (Lipljin, 2004, str. 111). Često je u programu potrebno ponoviti neku radnju više puta. Kako bi se izbjeglo pisanje iste naredbe mnogo puta, postoji programska struktura petlje koja omogućava ponavljanje jedne ili više naredbi. Petlja može biti: bezuvjetna, pri čemu se izvršava unaprijed zadan broj puta, i uvjetna pri čemu broj ponavljanja petlje ovisi o postavljenom uvjetu. Primjer strukture petlje vidljiv je Slici 4.

5. Primjena Scratcha u edukaciji

Scratch pridonosi razvoju mnogih vještina: učenja, kritičkog razmišljanja, rješavanja problema, komunikacije, suradnje, kreativnosti i inovacija. Te vještine uključuju jasno komuniciranje, analiziranje, sustavno i učinkovito surađivanje, kreativno razmišljanje te kontinuirano učenje. Ovakav pristup učenju i poučavanju inspiriran je konstruktivističkom teorijom učenja i obrazovanja (učenje temeljeno na iskustvu).

Scratch može poslužiti učiteljima, mentorima, nastavnicima pa čak i roditeljima kako bi napravili interaktivne programe, kvizove, zadatke te priče koje bi mogle biti usmjerene podučavanju raznih vještina, ali i školskih predmeta. Samim time bi mogao i poslužiti za ispitivanje znanja uz zanimljiv sadržaj. Scratch ne mora biti samo programski jezik s kojim će se djeca upoznat i koristiti ga samo na informatici ili izvanškolskim aktivnostima.

Scratch može poslužiti i za interaktivne, multimedijalne te animirane prezentacije za bilo koji predmet u osnovnoj školi. (primjer) interaktivna prezentacija zbrajanja, oduzimanja, množenja i dijeljenja brojeva uz kviz/igru za matematiku u nižim razredima osnovne škole.

Glazba je duboko ukorijenjena u čovjeku i sastavni je dio svih kultura svijeta te je važna za kvalitetan, skladan i cjelovit razvoj svakog pojedinca. Uporabom programskog jezika Scratch, učenici se mogu na zanimljiv način upoznati s izgledom i zvukom različitih instrumenata, osnovnim pojmovima (metar, tempo, dinamika, melodija...) mogu gledati i analizirati stvorene spotove/animacije te rješavati interaktivne glazbene kvizove.

Mogućnost korištenja Scratcha je sama mašta te samim time ima velik potencijal u edukaciji i razvoju apstraktnog razmišljanja kod djece.

6. Zaključak

S obzirom da su današnja djeca od malih nogu okružena tehnologijom, potrebno je iskoristiti njihov interes i podignuti razinu logičkog i apstraktnog razmišljanja, a učenje programskih jezika upravo je idealna platforma za ostvarenje toga cilja.

Izbor jezika koji se koristi u početnom programiranju uzrok je mnogih rasprava jer upravo taj „prvi“ jezik ima značajan utjecaj na programera. Osnovni cilj svakog uvodnog predmeta programiranja, a time i učenja prvog programskog jezika treba biti usvajanje osnovnih koncepata programiranja, ali mora biti intuitivan kako učenik ne bi odustao u samom početku.

Početni programi trebali bi biti vizualni tako da se jasno vidi način spajanja naredbi, zanimljivi i jednostavni za korištenje. Upravo takav je programski jezik Scratch koji omogućava jednostavno stvaranje interaktivnih priča, igara i animacija te dijeljenje radova preko interneta. Ovaj program idealan je za upoznavanje sa svijetom programiranja na poticajan način, bez kompliciranih izraza jer omogućuje shvaćanje logike programiranja bez upoznavanja sintakse i pravila programskih jezika.

Učenje programiranja pomoću Scratcha potiče djecu na razmišljanje, kreiranje te oživljavanje i realiziranje svojih ideja. Ono djeci daje samopouzdanje i vjeru kako mogu postati dizajneri i stvaratelji te ih uči međusobnoj suradnji. Također, učitelji bi svojim znanjem i entuzijazmom trebali kvalitetno obrazovati i motivirati učenike, jer programiranje kao digitalna vještina predstavlja novu pismenost koju je potrebno učiti od malih nogu baš kao učenje čitanja, pisanja i računanja.

LITERATURA

1. Bubica, N., Mladenović, M., Boljat, I. (2013.) Programiranje kao alat za razvoj apstraktnog mišljenja, (str. 1-11),, https://bib.irb.hr/datoteka/702093.Programiranje_kao_alat_za_razvoj_apstraktnog_miljenja-CUC-zbornik.pdf (21. svibnja 2019.)
2. Bubica, N. (2014). Strategije poučavanja i faktori koji utječu na unapređenje znanja programera početnika, <http://www.pmfst.eu/wp-content/uploads/2014/06/Istaziva--ki-seminar1-Bubica.pdf>
3. Lipljin, N. (2004). Programiranje 1.

4. Mujačić, E. Scratch priručnik za polaznike. Centar za održivi razvoj, http://www.cor.ba/publikacije/brosure/Scratch_prirucnik.pdf (20. svibnja 2019.)
5. Službena stranica Scratch editora, <https://scratch.mit.edu/projects/editor/?tutorial=home> (24. svibnja 2019.)

Summary

SCRATCH PROGRAMMING LANGUAGE - APPLICATION IN EDUCATION

Computer technology is spreading at an unstoppable pace into every part of human life and work. Today, children are getting familiar with new technologies in a different way than adults, because they are growing up with the new challenges. The role of the teacher is to direct students to use information technology creatively and, thus, raise the level of education in general. Informatics is an ideal area through which children can develop the skills necessary for lifelong learning, such as logical, creative and abstract thinking, creativity, learning by trial and error, upgrading and perfecting their own solutions continuously and using different sources of knowledge. Knowledge and ability to use essential tools and related technology efficiently, are today's must have skills, as well as possessing a range of skills covering levels from basic programming techniques to advanced problem solving. While today's children are tomorrow's adults and may not have to be programmers for a living, they will get practical knowledge on how the technology they use actually works and the indicative understanding of what can be achieved with it. One of the tools for initial learning of programming is Scratch, a programming language and development environment for the creation of stories, games, animation, music and other interactive content on the computer, specifically designed to be accessible and interesting to children. This paper will explain how to build applications by programming in Scratch, and the simple examples will be shown through the program structure which consists mainly of graphical blocks that represent commands.

Keywords: Scratch, programming, abstract thinking.

