

Učenje programiranja s programom Scratch

Learning Programming with Scratch

Sonja Lajovic
OŠ Kolezija
sonja.lajovic@gmail.com

Povzetek

Učitelji računalništva se velikokrat sprašujemo, kateri programski jezik je najprimernejši za učenje programiranja v osnovni šoli. Od leta 2007, ko je bil narejen programski jezik Scratch, je odgovor na to vprašanje precej lažji. Scratch je nastal namensko, prav za učenje programiranja. Otroci lahko z njim programirajo lastne računalniške igre, izdelujejo animacije ali interaktivne zgodbe. Program je novejši, zato ni na voljo dovolj literature o tem, na kakšen način poučevati, kako združiti prijetno s koristnim, da bi se otroci naučili osnovnih pravil računalniškega programiranja, hkrati pa delali računalniške igre, kar večino pri delu z računalnikom najbolj veseli. V prispevku bodo predstavljeni možni načini poučevanja programiranja s programom Scratch. Eden je poučevanje s programiranjem iger, drugi način pa je bolj podoben klasičnemu, sistematičnemu učenju posameznih programskih konceptov, podobno kot pri programu Logo. Prikazanih bo nekaj primerjav z drugimi programskimi jeziki ter izkušnje s poučevanjem Scratcha. Na kratko bodo predstavljene tudi osnovne značilnosti programa Scratch.

Ključne besede: Scratch, učenje, programiranje, igre, naloge

Abstract

IT teachers often ask themselves which programming language to use when teaching programming in primary school. The programming language Scratch, which started to be used in 2007, has helped to answer the question. Scratch was created to be used to teach programming. Children can use it to create their own computer games; they can make animations or interactive stories. Because the programming language is relatively new, not much has been written about how to use it effectively, so that the children would, on the one hand, learn the basics of programming and, on the other hand, learn to create their own games, which they usually enjoy the most. This article presents different ways of how the program Scratch can be used. One can use it for programming games but also in a more traditional way – to

systematically learn concepts pertaining to programming like in the programming language Logo. In this article, Scratch will be compared to some other programming languages; furthermore, examples of its usage in teaching programming will be presented. Main characteristics of Scratch will also be presented.

Keywords: Scratch, teaching, programming, games, exercise

1 Uvod

Kot učiteljica računalništva na osnovni šoli sem pri računalniškem krožku in pri izbirnih predmetih računalništva poučevala različne programske jezike, kot sta na primer Pascal in Logo. Pri tem sem naletela na različne težave, saj učenje klasičnih programskih jezikov, s katerimi lahko osvojiš pomembne elemente programiranja, zahteva veliko discipline in časa, preden prideš do rezultatov, ki so otrokom zanimivi in razumljivi. Tudi pričakovanja otrok, ki se vpišejo na računalniški krožek, so pogosto drugačna od tistih, ki se vpišejo na primer na logiko. Od leta 2007 sem začela pri računalniških predmetih uvajati program Scratch. Naredili smo nekaj računalniških igrice. Učenci so pokazali veliko zanimanja za tovrstno programiranje, vendar pa so se pojavile druge težave. Pokazalo se je, da primanjkuje literature o tem, kako poučevati Scratch. Na svetovnem spletu se sicer najde kar nekaj idej, predvsem v smislu kaj vse se da narediti s programom Scratch, manj pa takih, kako ga poučevati. Zato sem začela premišljevati o načinu poučevanja tega programa.

2 Možnosti poučevanja programa Scratch

2.1 Na kratko o Scratchu

Scratch je eden novejših (2007) programskih jezikov, ki je bil narejen prav za učenje programiranja. Otroci lahko z njim izdelujejo lastne računalniške igre, animacije ali interaktivne zgodbe. Svoje izdelke lahko prikazujejo na svetovnem spletu in jih delijo z drugimi. Program je razvil inštitut MIT Media Lab, pod vodstvom Mitchela Resnicka. Avtorji so se pri svojem delu zgledovali po znamenitih LEGO kockah.

Glavni elementi Scratcha so bitja in predmeti, ki jim v angleščini pravimo sprite, ustrezen slovenski prevod pa je figura. Figuro lahko otroci narišejo ali pa uporabijo katerekoli slike z računalnika. Figuri lahko dajejo navodila, kako naj se premika, kakšno glasbo naj predvaja, kakšen naj bo njen videz. Torej lahko otroci vsaki figuri napišejo program.

2.2 Poučevanje s pomočjo programiranja računalniških iger

Izkušnje, ki sem si jih nabrala pri poučevanju kažejo, da le redke otroke zanima, kako se programira, precej več pa je takšnih, ki bi se radi naučili izdelovati računalniške igre. Pokazalo se je, da je poučevanje osnov programiranja lahko za otroke precej bolj zanimivo in učinkovito, če se ga poveže z izdelovanjem računalniških iger. Otroci so pri takšnem delu bistveno bolj motivirani in ker je motivacija eden bistvenih elementov pri poučevanju, je smiselno začeti prav s programiranjem preprostih računalniških iger. Učenci s pomočjo učnih listov, ali s pomočjo razlage, začnejo izdelovati računalniško igro. Primer je igra akvarij, ki je prikazana na sliki 1.



Slika 1: Igra Akvarij

Glavni junak, rak, lovi ribe, ki se naključno gibajo po prostoru. Ko rak ujame ribo, ta za nekaj trenutkov izgine. Pri tej igri učenci spoznajo naslednje programske koncepte: zaporedje ukazov, zanke (*za vedno*), pogojni stavek (*če*), naključna števila, odziv na dogodke. Kot samostojno nalogo lahko učenci nato igri dodajo števec, ki šteje kolikokrat je rak ujel ribo. Pomagajo si s spremenljivko. Ko učenci ustvarijo novo spremenljivko, se prikažejo ukazi, ki so na voljo. Glede na izkušnje, večina učencev brez dodatne razlage ve, kako jih pravilno uporabiti.



Slika 2: Spremenljivka Rezultat

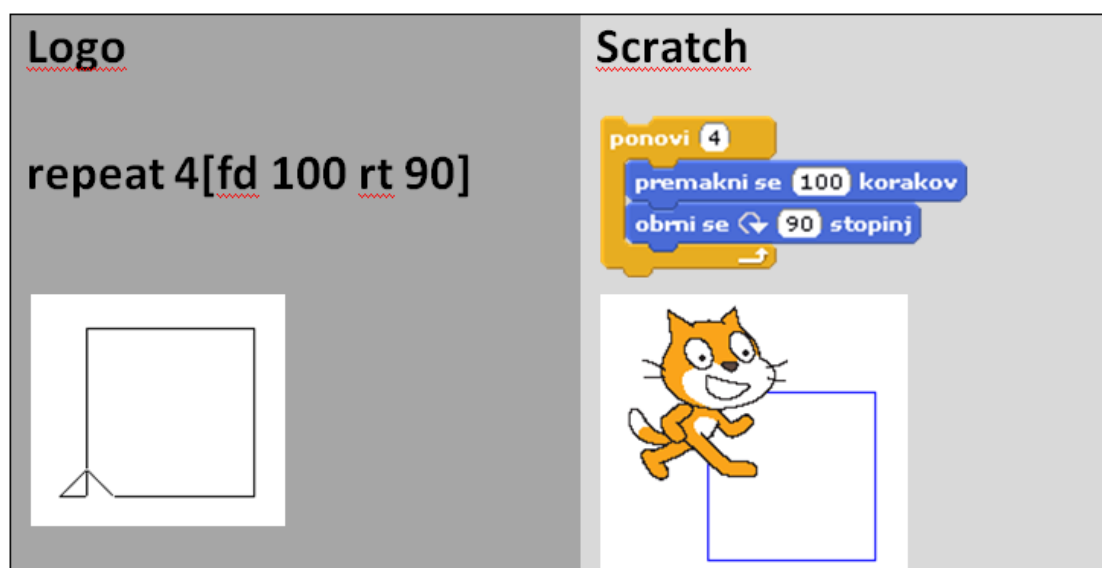
Igri lahko nato dodajo še hudobno ribo, ki se premika proti raku, torej ne naključno. Pri takšni nalogi lahko pride prav kakšen namig, saj učenci na začetku še ne poznajo dobro vseh ukazov, ki jih imajo na voljo. Na primer: pomagaj si z ukazom **obni se proti** **Rak**. Lažji igri sledi težja igra.

Učenje s programiranjem iger je zelo intuitivno, učenci pa so tudi zelo motivirani. Veliko učencev ima raje raziskovanje in praktično delo, kot pa učenje na klasičen način. Pozitivno je tudi, da se v heterogeni skupini lahko vključi in animira vse učence. Nekateri se bolj posvetijo programiranju, drugi risanju junakov, tretji razpredanju svoje domišljije. Slabo pa je, da je takšno pridobivanje znanja manj sistematično, težje je osvojiti vse koncepte. Včasih je tudi preveč intuitivno, zato se učenci velikokrat ne znajdejo, ko pride do napak v programu.

2.3 Klasično poučevanje z zbirko nalog, s pomočjo ukazov *svinčnik spuščen, šampiljka in reci*

Če v Scratchu uporabimo ukaz svinčnik spuščen, figura po naših navodilih riše črte in lahko nastajajo zanimivi vzorci. S pomočjo slike, ki nastane, lahko sproti preverjamo, ali smo prav razmišljali. Podobno kot pri programu Logo, kjer dajemo navodila želvi. Tako lahko naredimo zbirko nalog, ki sistematično obravnava določene koncepte programiranja. Taka zbirka nalog bo podobna tistim, ki obravnavajo Logo.

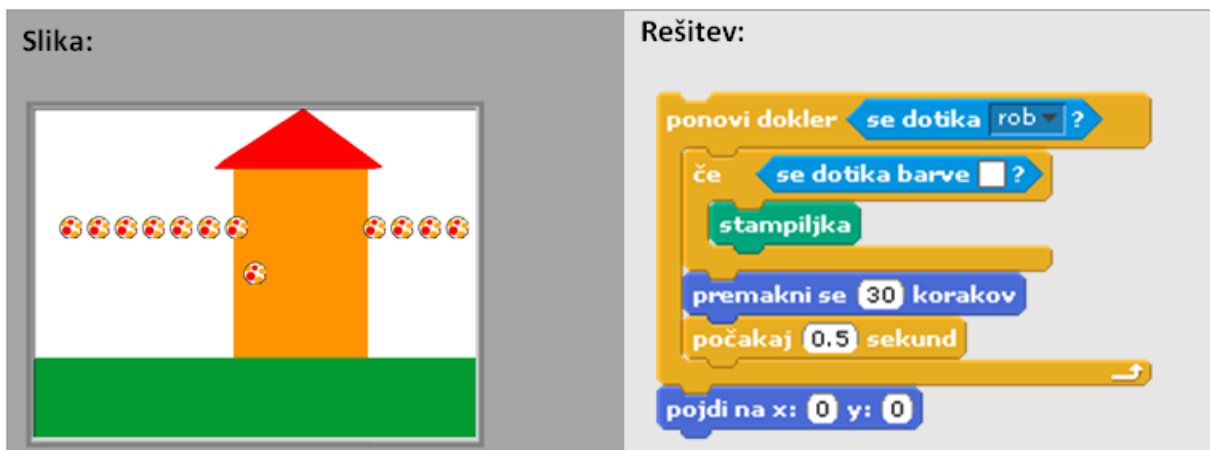
Primer naloge je prikazan na sliki 3. Na sliki je vidna tudi primerjava med pisanjem programa v Logu in v Scratchu. Kot opazimo, pri Scratchu namesto pisanja zapletenih programskih kod premikamo in zlagamo pisane grafične bloke, ki vsebujejo različne programske ukaze. Sestavljanje namesto pisanja prinese še eno prednost, odpade za marsikoga nepriljubljeno ukvarjanje s programsko slovnico. Prednost Programa Logo pa je, da podpira več programskih konceptov kot Scratch. Na primer procedure in funkcije, rekurzije, dedovanje, branje in pisanje iz datoteke.



Slika 3: Primerjava programske kode napisane v Logu oziroma v Scratchu

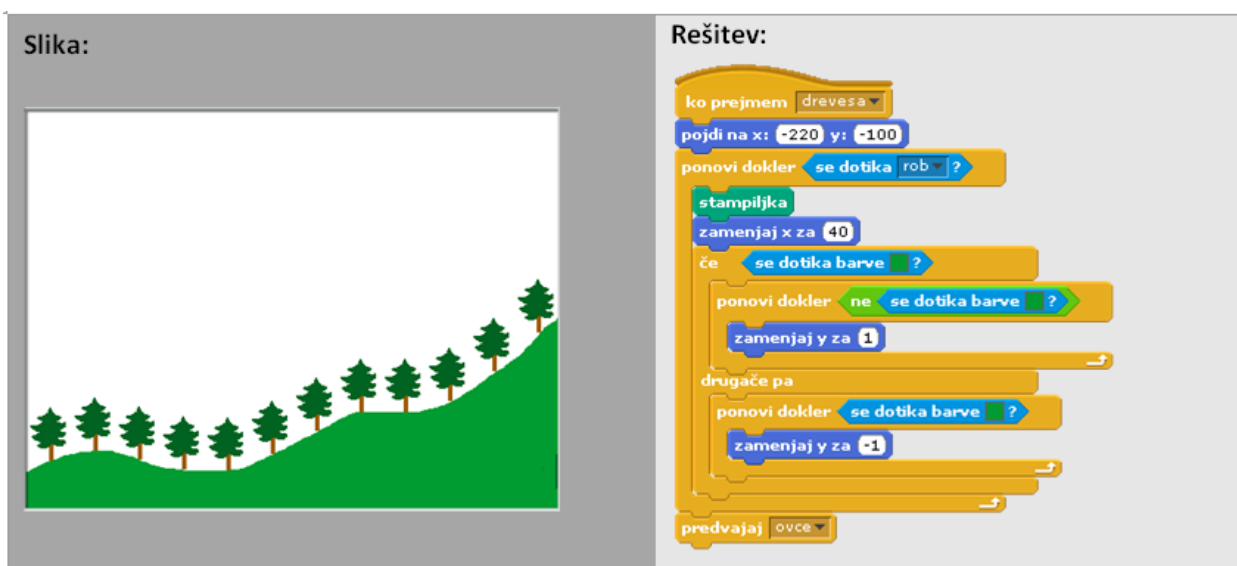
Drug način klasičnega učenja je programiranje s pomočjo ukaza *šampiljka*. Namesto črt nastajajo vzorci z različnimi motivi, ki so odtisi izbrane figure. Na spodnjih dveh primerih je prikazana ideja, kako sestavljati naloge.

1. primer: Sestavi program tako, da bo žoga puščala odtise na beli površini, dokler ne pride do konca odra. Ko pride do konca odra, se postavi točno v sredino.



Slika 4: Primer programa, ki naredi odtise žog

2. primer: Napiši program, ki na vrhu travnika nariše smreke.



Slika 5: Primer programa, ki na vrhu travnika nariše smreke

Tretji način je nekoliko drugačen, in sicer figura s pomočjo ukaza *reci* v oblčku pove, kar smo sprogramirali. To kar pove, lahko primerjamo z izhodnimi podatki. Na spodnji sliki je prikazan primer, ko maček pove poštrevanko števila 3.



Slika 6: Zaporedje ukazov gre sedmič v zanko in maček reče 21

Na prikazane tri načine se da pripraviti veliko zanimivih nalog, otroke pa animirajo slike, ki so jih ustvarili in govoreči maček. Večina otrok ohrani zanimanje, dokler so naloge lažje, pri težjih nalogah pa je potrebno več motivacije, vztrajnosti in sistematičnega dela pri spoznavanju konceptov. Vse to pa so že elementi klasičnega učenja programiranja, ki se oddaljuje od osnovnega namena Scratcha, to je programiranja igrice. Idealna kombinacija za uspešno učenje programiranja je torej nekje vmes, med klasičnim poučevanjem konceptov in bolj sproščenim učenjem skozi programiranje iger (Learning by Doing), učitelji pa morajo sami najti pravo razmerje, ki je vsakokrat posebno, odvisno od znanja in vneme otrok, ki se vpišejo v računalniški krožek.

3 Odzivi otrok na programiranje v Scratchu

Zaradi preproste uporabe lahko s Scratchem programirajo že otroci od 8. leta naprej. Na OŠ Kolezija smo uvedli programiranje s Scratchem v 4. razredu pri računalniškem krožku. Kljub temu, da so nekatere strukture Scratcha za otroke te starosti še zelo zapletene, so učenci izredno motivirani. Pri Scratchu jim je vseč prav vse: ustvarjanje figur po lastni želji, vključevanje multimedije in sestavljanje ukaznih blokov, torej programiranje. Zelo vseč ji je tudi možnost povezave Scratcha s kompletom kock Lego - WeDo. Kljub temu, da se radi zapletajo v za njih prezahtevnih računalniških programih, zlepa ne izgubijo interesa ter poskušajo in eksperimentirajo v nedogled. Torej nas učitelje čaka še en izziv, kako pripraviti primerno zbirko nalog in problemov za to starostno skupino torej za prve korake v programiranje.

Na Koleziji uvajamo Scratchu tudi pri izbirnih predmetih iz računalništva, v 7. In v 9. razredu. Tudi pri tej starosti se kaže veliko navdušenje za program Scratch. Navdušujejo jih podobne stvari kot mlajše otroke. Učenci te starosti, še posebej tisti iz 9. razreda lahko že zelo dobro razumejo vse strukture in elemente Scratcha. Še vedno pa je potrebna velika mera potrpežljivosti, da pridejo do zelenih rezultatov, kot je izdelava zahtevnejše igrice. Te pa včasih zmanjka, zato bi verjetno prišlo prav nekaj zunanje motivacije, kot so tekmovanja ali dogodki kot je npr. v tujini »Scratch day«.

4 Zaključek

Moje izkušnje pri uvajanju Scratcha na osnovni šoli Kolezija so pokazale, da je program Scratch zelo primeren programski jezik za učenje osnov računalniškega programiranja otrok v osnovni šoli. Možnih načinov in pristopov je več. V tem članku je bilo prikazanih nekaj možnosti.

Glede na razvoj znanja in potrebe računalniškega opismenjevanja se je pokazalo, da bi računalniško programiranje kazalo uvrstiti med samostojne izbirne računalniške predmete.

Viri in literatura:

- Juvan, M., Lokar, M. (1997): 121 nalog iz pascala, Društvo matematikov, fizikov in astronomov slovenije, Ljubljana.
- Gerlič, I.(1988): Prvi koraki v logo; ZOTKS, Ljubljana.
- Resnick, M. (2009), Scratch: Programming for all, Communication of the ACM, Vol. 52, No. 11, str. 60-67.