

ENAINDVAJSET OPORNIH TOČK ZA DELO S PROGRAMOM SCRATCH V UČILNICI ENAINDVAJSETEGA STOLETJA

Avtor: izr. prof. Hideki Mori¹, TITECH – Tokyo Institute of Technology

Prevajalec: Sašo Božič, spec. organizacije in managementa, OŠ Mengeš



¹ Hideki Mori. Twenty-One Things to Do with Scratch in the Twenty-First Century Classroom. Mednarodna konferenca Scratch2017BDX : Opening, Inspiring, Connecting, ki je potekala med 18. in 21. julijem 2017 v Bordeauxu v Franciji. Dostopno na <https://hal.inria.fr/hal-01563286>.
– Slike oz. fotografije, objavljene v pričujočem prevedenem članku, so prevzete iz originalnega članka.

Učenje programiranja za otroke kot dela temeljnih znanj 21. stoletja je deležno široke pozornosti javnosti po celem svetu. Vrednost učenja programiranja ni samo v učenju kodiranja, temveč tudi v tem, da otrokom pomaga tudi pri učenju učenja. Vendar pa – za razliko od tradicionalnih razredov, kjer učitelj učence uči večinoma neposredno – učenje kodiranja programov, računalnikov, sestavljanja robotov in drugih naprav (torej "programiranja" v širšem smislu) potrebuje drugačen pristop. Da bi postalo učenje programiranja v osnovnih šolah čim uspešnejše, morajo učenci postati samouki, učitelj pa mora učence pri samostojnem učenju spodbujati in delovati bolj kot posrednik znanj, mentor ali sovrstnik (in občasno kot učitelj). Učilnice imajo številne fizične omejitve, šolski učni načrt in še posebej učni načrti za osnovne šole pa imajo omejitev še več, tako časovnih kot vsebinskih. Kako lahko torej uvedemo programiranje v učilnice 21. stoletja?

Knjiga *Twenty Things to Do with a Computer* (Papert and Solomon 1971) je ob izidu vplivala tako na učitelje kot na raziskovalce. V tem prispevku bom z vami delil enaindvajset pristopov k učenju, ki sem jih prakticiral pri poučevanju v japonskih osnovnih šolah skozi več kot 700 učnih ur v zadnjih petnajstih letih in s katerimi sem dvignil kvaliteto učenja programiranja pri delu s programom Scratch in majhnim računalnikom Cricket, ki ga lahko programiramo in ga je razvil MIT Media Lab.

21 opornih točk za delo s Scatchem in drugimi programskimi orodji v učilnici

Programiranje je za učence zelo zabavna dejavnost. Medtem ko je najpomembnejše, da se pri teh dejavnostih zabavajo, je potrebno vzpostaviti tudi okvire, s katerimi zagotovimo, da učenci ne preživijo svojega časa v razredu samo ob zabavi. Spodaj so opisane oporne točke, o katerih razmišljam, ko načrtujem in vodim razrede v osnovnih šolah.

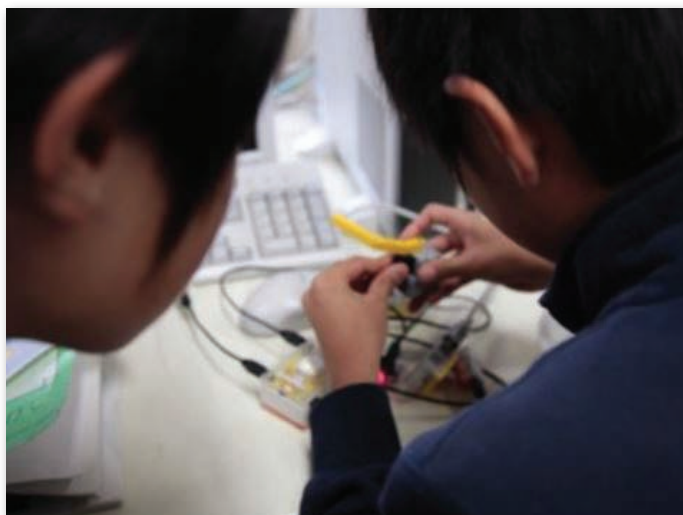
1. Postanite poklicni strokovnjaki

Pri učenju programiranja morajo učenci postati aktivni udeleženci izobraževanja. Če učence spodbujamo, da se vživijo v vloge poklicnih

strokovnjakov, kot so izumitelji, oblikovalci in raziskovalci, se lažje poistovetijo s cilji dejavnosti, ki jo izvajajo. Kadar se poistovetijo s cilji, ki jim nekaj pomenijo, se veliko enostavneje vrnejo na pravo pot, ko zaidejo med razmišljanjem in delom pri programiranju.

2. Učenci naj postanejo mali učitelji in učitelji veliki učenci

Spodbujanje učencev, da kot mali učitelji aktivno pomagajo drugim učencem, ki se soočajo s težavami pri programiranju, je učinkovit način vodenja pouka. Med poukom se pojavi skoraj toliko različnih problemov, kot je učencev. Učenci se veliko naučijo s



samostojnim reševanjem problemov, ko pa se soočijo s problemom, ki ga sami ne morejo rešiti, potrebujejo čim hitreje ustrezen nasvet nekoga, ki zna problem rešiti, pa naj bo to učitelj ali drugi učenci, ki poznajo rešitev problema. Ker lahko učitelj med šolsko uro pomaga le omejenemu številu učencev, lahko učenci, ki poučujejo druge učence, veliko pripomorejo k uspešnemu vodenju pouka.

3. Zapišite cilje učne ure

Spirala kreativnega učenja (Resnick 2007) je učinkovit način učenja programiranja. Za realizacijo procesa spirale kreativnega učenja pri časovno omejeni učni uri je potrebno razumeti aktivnosti "Predstavljaj si" - "Ustvari" - "Predvajaj" - "Deli" - "Razmisli". Če zapišemo, kakšen namen imajo te aktivnosti pri učni uri, lahko učenci zavestno usmerjajo svoje dejavnosti med učno uro in predvsem spodbujajo aktivnost "Predstavljaj si".

4. Učenci naj med seboj sodelujejo

Izvajanje aktivnosti v sodelovanju z drugimi učenci je eden izmed temeljnih pristopov konstruktivističnega učenja, ki ustvarja obilo možnosti za učenje programiranja. Učitelj s spodbujanjem samostojnega dela, poleg spodbujanja sodelovanja v skupini, prepreči, da bi se učenci oprli samo na delo skupine in ne bi ničesar ustvarili samostojno. Učenci lahko tako sodelujejo pri učinkoviti interakciji med samostojnim in skupinskim učenjem.

5. Razmislite o delu, ki ste ga opravili pri učni uri

Razmislek o delu, ki smo ga opravili med učno uro, naj poteka v zadnjem delu učne ure in naj bo razdeljen na dva sklopa. V prvem sklopu naj učenci razmišljajo o tem, kaj so delali pri tej učni uri, in svoje misli povzamejo in zapišejo v zvezku tako, da bodo lahko svoje ugotovitve delili z razredom. V drugem sklopu naj učenci poslušajo ugotovitve sošolcev in po tem ponovno razmislijo in dopolnijo svoja razmišljanja, v katera naj vključijo svoje ideje o tem, kaj želijo preizkusiti naslednjo uro.

6. Izmenjajte odkritja, do katerih se učenci dokopljejo med učno uro

Če želite, da učenci medsebojno delijo tisto, kar so odkrili med učno uro, jim kratek čas za predstavitev svojih ugotovitev ob koncu ure zadostuje. Če pa jim želimo ponuditi priložnosti za neformalno izmenjavo mnenj že med samo učno uro, je za učence pomembno, da jim dovolimo hoditi naokrog po učilnici, kadar odkrijejo kaj zanimivega in bi želeli to deliti z drugimi. Učenci so vedno aktivni pri neformalnem izmenjevanju idej, zato lahko pogosto ugotovimo, da ustvarjajo podobne projekte kot učenci v njihovi bližini.

7. Bodite dobri posnemovalci

Učenci dostikrat ne želijo posnemati idej drugih ali dovoliti, da drugi posnemajo njihove. Učitelj jim mora pojasniti, da lahko posnemajo dobre

ideje in jih aktivno vključijo v svoje delo. Pri tem mora učitelj hkrati spodbujati učence k razvijanju lastnih idej in jih učiti spoštovati delo učencev, ki so ustvarili originalne ideje.

8. Predstavite svoje projekte drugim učencem

Pomembno si je vzeti čas za predstavitev učenčevih projektov drugim učencem. Če poteka učna ura programiranja v več različnih razredih, je zaželeno, da imajo učenci, poleg kratkih predstavitev v svojem razredu, dovolj časa tudi za pripravo predstavitev svojega dela v ostalih razredih. S povzemanjem svojih misli v predstavitvi projekta in pridobivanju mnenj drugih učencev dobijo učenci dostikrat nove ideje.

9. Naredite samoevalvacijo

Za razliko od drugih učnih predmetov je pri učencih, ki delajo na lastnih projektih programiranja, težko določiti enotne učne cilje. Zato je pomembno, da si učenec sam zastavi svoje cilje, ki bi jih rad dosegel, in potem tudi v samoevalvaciji sam oceni svoje dosežke.

10. Poskrbite, da napake štejejo

Iskanje napak v programu je najmočnejše orodje za učenje programiranja (Papers 1980). Večina učencev se nerada sooča s svojimi napakami. Učitelj mora učencem dopovedati, da je delanje napak pri pouku programiranja lahko dobro, pri čemer je najpomembnejše, da napake odkrijejo in potem sami poiščejo rešitve zanje.





11. Dajte učencem priložnost za ponovni poskus

Če želimo omogočiti učencem, da bi se učili s poskusi in napakami, mora učitelj dati učencem na voljo več časa za samostojno programiranje in ustvarjanje. V nekaterih primerih sploh ni mogoče preizkusiti novih idej, če ne spremenimo samih temeljev projekta in začnemo od začetka. Ko učencem omogočimo, da večkrat delajo na istem projektu, le-ti vanj lažje vključijo izkušnje iz prejšnjih poskusov in uporabijo ideje, ki so jih v vmesnem času dobili od drugih učencev.

12. Učenci naj pomagajo pri pripravi učilnice

Učenci bodo verjetno aktivneje sodelovali pri pouku, če bodo sami pripravili pripomočke za delo, namesto da bi jih vse preprosto pričakalo pripravljeno v razredu. Smiselno je prositi učence, naj tudi doma zbirajo material za izdelavo projektov in ga prinesejo v šolo. Prav tako je lahko koristno, če sami določimo učence, ki

bodo pripravili razred skupaj z učitelji (čeprav potem včasih traja dlje časa, kot če bi učitelj razred pripravil sam).

13. Učenci naj pospravijo učilnico

Učitelj naj ne bi sam pospravil učilnice po učnih urah, pri katerih se je uporabljalo veliko pripomočkov in materialov, kot je na primer izdelovanje robotov. Ko učenci sami pospravljajo in razstavljajo izdelke, ki so jih naredili, dobijo priložnost, da ponovno razmislijo o postopku izdelave.

14. Pritegnite pozornost razreda

Ko so učenci osredotočeni na programiranje, pogosto težko preusmerijo svojo pozornost na druge dejavnosti, kot je na primer poslušanje učitelja. Smiselno je, da učitelj od učencev izrecno zahteva pozornost, ko spregovori. Učenci bodo verjetno kljub temu težko preusmerili pozornost, zato mora učitelj razumeti, kako zelo so učenci osredotočeni pri programiranju, namesto da jim samo ukaže

poslušanje kot kakšnemu računalniškemu programu.

15. Bodite učitelj, ki ne nauči vsega

Učitelj pri učenju programiranja ne more učencev naučiti vsega, še posebej kadar pri pouku uvaja uporabo novih tehnologij in programov. Ko se učitelj zavestno odpove želji, da bi učence naučil vse, lahko začne skupaj z učenci raziskovati nove neznane stvari, hkrati pa se osvobodi strahu pred novimi didaktičnimi gradivi.

16. Ocenite predznanje otrok

Na temo računalništva učenci pridobijo že veliko sorodnih znanj drugje, zato si je težko ustvariti sliko o njihovem osnovnem predznanju in sposobnostih. Programiranje nam ponuja priložnost, da preverimo, kako učenci rešujejo izzive, in opazujemo, kako pristopijo k učenju. (V zadnjem času obstaja vedno več ustanov izven šole, kjer se učenci učijo programiranja in sestavljanja robotov, zato je potrebno biti pri oceni še bolj pazljiv.)



17. Pustite učencem, da imajo nekaj časa zase

Učence zanimajo zelo različne stvari. Tudi če se učitelju zdi, da so učencevi interesi daleč od učiteljevih ciljev, mora učenca voditi v pravi smeri. Da učencev ne bi neprestano usmerjali in jih s tem preveč omejevali, jim moramo pustiti, da imajo nekaj časa zase in svoje ideje. Tehnične težave, ki se pojavijo pri delu učencev, pa je potrebno reševati sproti.

18. Učenci naj razmišljajo s celim telesom

Ko učenci programirajo, razmišljajo samo z možgani. Pomembno pa je, da se učenci med učenjem oz. učno uro (Papert 1980) naučijo poleg svojih možganov uporabljati tudi svoje telo. To jim najbolje predstavimo z vajo, kjer prevzamejo vlogo robota in premikajo svoje telo v skladu z ukazi, kot je na primer "dvigni roko", ko učitelj (program) zahteva, da to storijo.

19. Pogovorite se o programih in računalnikih, ki so učencem blizu

Računalniki so vgrajeni v veliko stvari, ki obdajajo učence, kot so aparati, igrače in video igre. Vse te stvari nadzirajo in upravljajo računalniški programi. Če učenci razmišljajo o tem, kako znane stvari delujejo in kako bi lahko te stvari sami izdelali in sprogramirali, bodo s tem poglobili svoje razumevanje računalnikov in programiranja.

20. Povejte učencem, da se skozi programiranje učijo, kako se učiti

Veliko učencev, staršev in učiteljev še vedno meni, da je učenje programiranja potrebno samo za obvladovanje znanj programiranja (kodiranja). (Naj omenim en tak primer: Deklica prvega razreda mi je povedala, da znanja programiranja ne potrebuje, saj želi biti cvetličarka.) Učitelj mora učencem povedati, da je programiranje in sestavljanje robotov ter drugih naprav zelo koristno tudi za učenje o tem, kako se pravzaprav učiti

21. Razmislite o tem, kako uporabiti izkušnje pri drugih učnih urah

Tako učenci kot učitelji lahko uporabijo svoje izkušnje in pristope iz učenja programiranja tudi pri pouku drugih predmetov. Razmišljanje o tem, kako bi učenci in učitelji lahko uporabili pridobljene izkušnje, lahko vodi k uvajanju novih učnih praks tudi pri drugih šolskih predmetih. S tem lahko dodamo še en nov kamenček v mozaik izobraževanja v učilnici nove generacije (učilnici 21. stoletja).

Zahvale

Rad bi se zahvalil vsem učencem in učiteljem, ki delajo z mano v razredu. Še posebej bi se rad zahvalil Manabu Sugisawi, saj večina idej, ki so predstavljene v tem prispevku, izhaja iz sodelovanja v njegovem razredu. To delo je podprlo JSPS KAKENHI (številka odobritve: 17K01110).

Literatura

Kafai, Y. and Resnick, M. (1996) *Constructionism in Practice*. Lawrence Erlbaum Associates, New Jersey.

Papert, S. and Solomon, C. (1971) *Twenty Things to Do with a Computer*, Artificial Intelligence Memo No. 248, A.I. Laboratory, Massachusetts Institute of Technology.

Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.

Resnick, M. (2007) All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. ACM Creativity & Cognition Conference, Washington DC, June 2007.

OPOMBA PREVAJALCA

Veliko držav še danes v primarno izobraževanje otrok na nacionalni ravni nima integriranega učenja programiranja. V Sloveniji smo s sistematičnim poučevanjem programiranja v osnovni šoli začeli v šolskem letu 2014/2015, in sicer v drugi triadi osnovne šole pri pouku neobveznega izbirnega predmeta Računalništvo. V tem času se je nabralo kar nekaj izkušenj, s katerimi bi bilo smiselno obogatiti prevod članka *Twenty-One Things to Do with Scratch in the Twenty-First Century Classroom* – Enaindvajset opornih točk za delo s programom Scratch v učilnici enaindvajsetega stoletja.



Učenje programiranja otrok je za pedagoga svojevrsten izziv. Otroci, ki se v slovenskih šolah lahko vključijo v učenje programiranja pri neobveznem izbirnem predmetu Računalništvo, so stari med 9 in 12 let. Po normativu je lahko v razredu pri eni učni uri hkrati največ 24 učencev. Njihova starost, učne sposobnosti in predznanje so zelo različni. Poučevanje te raznolike populacije je mogoče le z zelo visoko stopnjo učne diferenciacije in individualizacije pouka. To zahteva od učitelja veliko predpriprav na učno uro, polno angažiranost med izvajanjem pouka ter predvsem neizmerno stopnjo potrpežljivosti.

Zelo zahtevno delo učitelju olajša visoka stopnja motivacije in navdušenja učencev nad programiranjem. Učenci se veselijo dela z računalnikom, dodatno pa jih k delu spodbudi misel, da bodo izdelali igrico. Igranje iger jih tako močno pritegne, da so navdušeni že nad najbolj preprosto igro, ki jo sami sprogramirajo. Igranje svoje igrice, ki pravilno deluje, dojemajo kot nagrado za uspešno opravljeno delo. Učitelj mora pri načrtovanju dela paziti, da je koncept programiranja pri vsaki igrici vsaj malce drugačen in zahtevnejši od prejšnjega. S tem postavi pred učenca vsakič nov izziv in ga spodbudi k iskanju novih rešitev in pristopov. Učenci so pri svojem delu bolj zavzeti, če jih aktivno vključimo v pripravo idejne zasnove igrice. Razmišljanje o scenariju, junakih in zlobnežih, zbiranju točk ter izgubljanju življenj, skozi

katero spoznavajo celotni proces razvoja igre, hkrati razvija njihovo domišljijo in smisel za načrtovanje dela. V idejah največkrat presežejo svoje sposobnosti programiranja, zato mora učitelj v tem možganskem viharju večkrat preusmeriti njihovo razmišljanje v dosegljive okvire.

Pri poučevanju moramo upoštevati, da se pravila programiranja in računalniškega razmišljanja v nekaterih segmentih zelo razlikujejo od razmišljanja ljudi. Učitelj mora zato vložiti precej truda v učenje novih miselnih vzorcev in procesov pri učencih. Učenje programiranja ni mogoče, če učenec že od samega začetka ne razume razlik med človeškim in računalniškim razmišljanjem.

Večina otrok v tej starostni skupini težko razume abstraktne računalniške pojme. Pri poučevanju mora zato učitelj poiskati primere iz realnega sveta ali pa odigrati pred razredom vlogo neumnega robota ter s takim pristopom ponazoriti težko razumljive računalniške procese. Kljub takšnim aktivnostim se večkrat zgodi, da učenec ali skupina učencev, zaradi svojega načina dojetanja, razlage ne razume, in je potrebno isto vsebino razložiti ali jo ponazoriti na popolnoma drugačen način, z drugega zornega kota. Posamezni učenci osvojijo principe programiranja zelo hitro, nekateri potrebujejo več časa.

Med samostojnim delom učenci zelo neradi dvignejo pogled z zaslona in

preusmerijo pozornost na učitelja. Pri večji skupini otrok je to še toliko težje, saj prvi že spusti pogled nazaj na zaslon, ko ga zadnji učenec šele dvigne. Na naši šoli smo pri gradnji nove računalniške učilnice načrtovali učilnico z mislijo na ta problem. V novi učilnici imajo računalniški zasloni, ki jih uporabljajo učenci, ločeno električno napajanje, pri čemer je napajanje vseh zaslonov učencev vezano na stikalo pri učiteljevi mizi. Ko želi pozornost učencev, učitelj tako z enim stikalom preprosto ugasne vse zaslone učencev, pri čemer računalniki delujejo nemoteno naprej. Pozornost učencev je v trenutku pri učitelju, saj vedo, da bodo zasloni ostali ugasnjeni. Ko konča z razlago, zaslone vklopi nazaj in učenci lahko nadaljujejo svoje delo tam, kjer jih je prekinil učitelj.

Merljivi pozitivni učinki izvajanja programa učenja programiranja se bodo verjetno pokazali šele čez nekaj let skozi povečan vpis na tehnične smeri v sekundarnem in terciarnem izobraževanju ter posledično tudi skozi večjo razpoložljivost kakovostnega tehničnega kadra. Glavni cilj učenja programiranja v osnovi šoli je učencem približati računalniški jezik in razmišljanje. Ne vzgajamo množice bodočih programerjev, temveč populacijo, ki bo razumela delovanje računalniških programov in znala to svoje znanje vključiti in povezati z multidisciplinarnim poklicem prihodnosti, ki ga bo opravljala.

