

**UNIVERZA V LJUBLJANI**  
**PEDAGOŠKA FAKULTETA**

**ALEŠA ŽANDAR**

**IZDELAVA UČNIH PRIPRAV ZA POUČEVANJE**  
**PROGRAMIRANJA S SCRATCHEM**

**Diplomsko delo**

**Ljubljana, 2014**



**UNIVERZA V LJUBLJANI**  
**PEDAGOŠKA FAKULTETA**  
**UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE**  
**DVOPREDMETNI UČITELJ**  
**SMER: MATEMATIKA - RAČUNALNIŠTVO**

**ALEŠA ŽANDAR**

**Mentor: DOC. DR. IRENA NANČOVSKA ŠERBEC**

**Somentor: AS. MATEJ ZAPUŠEK**

**IZDELAVA UČNIH PRIPRAV ZA POUČEVANJE PROGRAMIRANJA  
S SCRATCHEM**

**DIPLOMSKO DELO**

**Ljubljana, 2014**



## **ZAHVALA**

*Zahvala gre moji družini in prijateljem, ki so mi potrpežljivo stali ob strani in me podpirali v času študija in pisanja diplomskega dela.*

*Posebna zahvala pa gre tudi mentorici doc. dr. Ireni Nančovski Šerbec in somentorju asist. Mateju Zapušku za čas, nasvete, ideje in strokovno pomoč pri izdelavi diplomskega dela.*



## **POVZETEK**

Dneva brez računalnika si danes težko predstavljamo. Da bi bili kos tako hitremu razvoju tehnologije pa ni dovolj, da le znamo uporabljati določena programska orodja. Težimo k temu, da bi otroci razumeli delovanje ter uporabo le-teh zamenjali s kreiranjem. V skladu s tem pa se spreminjajo tudi učni načrti računalniških predmetov. V šolskem letu 2014/15 se v drugo triletje osnovne šole vpeljuje neobvezni izbirni predmet Računalništvo. Namen diplomskega dela je z učnimi gradivi pomagati učiteljem pri uporabi vizualnega programskega okolja Scratch s katerim otroci začnejo programirati.

V prvem delu diplomskega dela bomo predstavili teoretične osnove kognitivnih psihologov. Na podlagi njihovih dognanj še danes oblikujemo učne ure v šolah in želimo spodbujati učenčevo ustvarjalnost in samostojno gradnjo znanja.

Nadaljevali bomo z opisom nekaterih vizualnih programskih orodij, ki spodbujajo razumevanje osnovnih računalniških konceptov. Predstavili bomo poučevanje računalniških vsebin, povezanih z novim učnim načrtom, preko vizualnega programskega orodja Scratch. Sledila bo realizacija ene izmed vsebin za poučevanje s Scratchem.

Za konec pa bomo opisali postavitev spletnega portala na katerem bodo objavljena učna gradiva in bo služil kot pomoč učiteljem, ki se bodo letos prvič srečali z novim neobveznih izbirnim predmetov Računalništvo.

**Ključne besede:** učna gradiva, neobvezni izbirni predmet Računalništvo, vizualna programska okolja, uvod v programiranje, Scratch

## **ACM klasifikacija:**

K.3 Računalništvo in izobraževanje

K.3.1 Uporaba računalnikov v izobraževanju

K.3.2 Poučevanje računalništva in informatike



## **ABSTRACT**

It is hard to imagine a day without the computer. To cope with such rapid development of technology, it is not enough to only know how to use certain software tools. We strive to make children understand the computer concepts and change the application with creation. In line with this, the curriculum is also changing. In the second trimester of the school year 2014/15 an optional Computer and Information Science subject is introduced. The purpose of the thesis is to assist teachers with the help of teaching materials by using visual programming environment Scratch, in which children start to program.

In the first part of the thesis we will present the theoretical basis of cognitive psychologists. Teachers still design lessons in schools based on cognitive psychologist research and we also want to promote students creativity and self-construction of knowledge, based on their research.

We will continue with a description of some visual programming tools that promote the understanding of basic computer concepts. We will present the teaching of computer-related content with the new curriculum through a visual programming tool Scratch. This will be followed by the realization of one of the teaching materials for Scratch

In the end we will describe the layout of the web portal, where we will publish learning materials and will serve as an aid to teachers who will teach new optional course Computer and Information Science for the first time.

**Key words:** learning materials, optional subject Computer and information Science, visual programming environment, introduction to programming, Scratch

## **ACM Computing Classification System**

### K.3 Computers and Education

#### K.3.1 Computer Uses in Education

#### K.3.2 Computer and Information Science Education

## KAZALO VSEBINE

ZAHVALA .....	5
POVZETEK .....	7
ABSTRACT .....	9
KAZALO VSEBINE.....	11
KAZALO SLIK.....	13
1. UVOD.....	1
1.1. Pregled vsebine ostalih poglavij .....	3
2. TEORETIČNE OSNOVE .....	4
2.1. Kognitivizem .....	4
2.2. Konstruktivizem.....	8
2.3. Razlika med konstruktivizmom in konstrukcionizmom .....	10
2.4. Uporaba Scratcha in konstrukcionizem .....	11
3. VIZUALNA PROGRAMSKA OKOLJA .....	12
3.1. Scratch .....	12
3.2. Alice 3D.....	14
3.3. Kodu.....	14
4. PROGRAMIRANJE V OSNOVNI ŠOLI.....	16
5. EMPIRIČNA RAZISKAVA .....	19
Analiza odgovorov.....	20
Ugotovitve .....	24
6. PRIPRAVA GRADIV .....	25
6.1. Pedagoška osnova .....	25
6.2. Tehnološko okolje – portal z gradivi .....	30
7. NADALJNJE DELO .....	32
8. ZAKLJUČEK .....	33
9. LITERATURA .....	34



## **KAZALO SLIK**

Slika 1: Scratch maček .....	12
Slika 2: ScratchJr .....	14
Slika 3: Alice logotip .....	14
Slika 4: Kodu logotip .....	14
Slika 5: Uporabniški vmesnik programa Kodu .....	15
Slika 6: BitVise SSH Client .....	30
Slika 7: BitVise SFTP okno .....	31
Slika 8: Konfiguracija okolja Wordpress .....	31



## 1. UVOD

Otroke, ki so se rodili, ko je digitalna tehnologija že obstajala, Prensky opiše kot digitalne domorodce (Prensky, 2001). Zanje je značilno, da tekoče obvladajo »jezik« računalnikov, videoiger, interneta ipd. Tako jim poučevanje računalništva v smislu učenja veščin uporabe programov ne predstavlja večjega problema.

V članku *Should everybody learn to code* je Mark Surman zapisal: »Koda je postala del opismenjevanja. Vsi moramo razumeti kako svet deluje, ne le inženirji.« Jeannette Wing pa je mnenja, da se morajo vsi naučiti algoritmičnega razmišljanja. Algoritmično razmišljanje nam omogoči in nas, med drugim, nauči abstraktnega mišljenja in tako smo sposobni problem razčleniti na manjše dele, ki jih nato lažje rešimo. Veščina reševanja problemov je zelo zaželenja tudi v pedagoškem svetu. Eden od načinov učenja teh veščin je programiranje, pravi Jeannette Wing, podpredsednica Microsoft Research in redna profesorica na univerzi Carnegie Mellon v ZDA. Po Futschek (2006) je algoritmično razmišljanje množica sposobnosti, ki so povezane z ustvarjanjem in razumevanjem algoritmov. V množico sposobnosti uvršča:

- sposobnost analize danega problema,
- sposobnost natančne specifikacije problema,
- sposobnost iskanja osnovnih korakov za rešitev problema,
- sposobnost z osnovnimi koraki izgraditi pravilen algoritem za dani problem,
- sposobnost razmišljanja o vseh možnih izvedbah algoritma in robnih pogojih in
- sposobnost izboljšanja učinkovitosti algoritma.

Ni tako pomembno, da otroke naučimo golega zapisovanja programske kode. Bolj pomembne in uporabne so veščine, ki nas pripeljejo do rešitve določenega problema. Naučiti jih moramo veščin reševanja problemov in kritičnega ter algoritmičnega mišljenja, trdi Jeannette Wing (*Should everybody learn to code*, 2014). Ni tako pomemben programski jezik, kot je pomembno razumevanje osnovnih konceptov reševanja problemov (*Computer programming goes back to school*, 2013). Torej je bolj smiselno otrokom ponuditi znanja, ki bi jim omogočala boljše razumevanje računalniških konceptov in jim dala vpogled v delovanje računalnikov, hkrati pa spodbujala kreativno razmišljanje in jim ponudila razvoj v smislu ustvarjanja novih vsebin.

Računalniška znanja pa niso uporabna zgolj za bodoče računalničarje, ampak tudi za ostale, saj so koristna tudi na drugih predmetnih področjih in razvijajo znanja in spretnosti za uspešno vključevanje posameznika v družbo. Logično sklepanje, strukturirano reševanje problemov, abstrakcija, dekompozicija, iteracija, rekurzija so le nekatere metode, veščine in znanja, ki jih učenci razvijajo pri reševanju računalniških problemov (Selby in Wooland, 2014). Danes večina učiteljev, ki poučujejo računalništvo v osnovni šoli, ne posveča pretirane pozornosti programiranju. Eden od razlogov je zagotovo to, da je ta v učnih načrtih za izbirne predmete označen z zvezdico in tako sodi med dodatne vsebine in ne med temeljna znanja (Stopar et al., 2013).

V prihodnjem šolskem letu se bo v drugem triletju osnovne šole začel izvajati neobvezni izbirni predmet Računalništvo. Učni načrt tega predmeta so na Ministrstvu za izobraževanje, znanost in šport sprejeli letos in ga uvrstili v razširjen program osnovnih šol. Predmet ne bo temeljil na učenju uporabe tehnologij, ampak na razumevanju njihovega delovanja. Sledil bo svetovnim trendom, ki stremijo k temu, da bi poučevanje uporabe računalnikov zamenjali s poučevanjem računalniških konceptov in računalniškega razmišljanja (Kranjc, et al., 2014). Takšno poučevanje pa zahteva več priprav, posebne učne materiale in ustrezno usposobljene učitelje.

Z vpeljavo novega neobveznega izbirnega predmeta Računalništvo v osnovne šole, bo potrebno učitelje dodatno izobraziti in podkrepiti z računalniškimi vsebinami, ki se do sedaj še niso poučevale. Predvidevamo, da bodo učitelji pri sestavi učnih materialov potrebovali pomoč. V ta namen bomo izdelali predlog učnega gradiva za poučevanje v vizualnem programskem okolju Scratch (Scratch, 2014).



## **1.1. Pregled vsebine ostalih poglavij**

V diplomskem delu je glavni cilj izdelati učna gradiva za nov neobvezni izbirni predmet Računalništvo, ki se z letošnjim šolskim letom vpeljuje v osnovne šole. V ta namen bomo v prvem delu diplomskega dela opisali učne teorije, ki nam omogočajo vpogled v otrokov razvoj in posledično razumevanje otrokovega razmišljanja in učenja. Pogledali si bomo kaj vpliva na otrokovo učenje, mišljenje in osmišljanje sveta okoli njega.

V nadaljevanju bomo predstavili nekatera vizualna programska okolja, ki so dobro orodje za učenje programiranja mlajših in s katerimi spodbujamo aktivno konstruiranje novih idej.

V zadnjem delu diplomskega dela bomo predstavili postavitev spletnega portala v okolju *WordPress*, namenjenega učiteljem, ki bodo poučevali nov neobvezni izbirni predmet Računalništvo. Na portalu bodo imeli možnost pregledovanja učnih predlogov in bodo dostopna vsem. Na spletnem portalu bo mogoče prenašanje učnih gradiv na računalnik in komentiranje že naloženih gradiv, ki bodo objavljena na fakultetnem strežniku *Hrast* (*hrast.pef.uni-lj.si/ris*).

Znotraj diplomskega dela bomo predstavili rezultate intervjujev, ki smo jih izvedli med osnovnošolskimi učitelji. Z intervjujem smo želeli ugotoviti, katere vsebine novega neobveznega izbirnega predmeta Računalništvo učiteljem predstavljajo največji izziv in kako jim je všeč ideja o izmenjavi gradiv in s pomočjo preko spletnega portala.

## 2. TEORETIČNE OSNOVE

V poglavju bomo opisali učne teorije znanih kognitivnih psihologov, na podlagi katerih še danes učitelji oblikujejo učne ure. Poglavje je povzeto po knjigi Pedagoška psihologija, avtorice Woolfolk Anite (Woolfolk, 2002). Zadnji del poglavja, *Razlika med konstruktivizmom in konstrukcionizmom*, je povzet po članku *Piaget's Constructivism, Papert' Constructionism: What's the difference?* avtorice Edith Ackermann (Ackermann, 2001) in po članku Seymour Papert's Legacy: *Thinking about learning, and learning about thinking*, avtorja Paulo Blikstein (Blikstein).

Poglavje bomo zaključili z razlago teoretičnih osnov učenja s Scratchem (Resnick, 2012).

### 2.1. Kognitivizem

Švicarski psiholog Jean Piaget se je ukvarjal s psihološkim razvojem od rojstva pa vse do odraslosti. V zadnji polovici 20. stoletja je Piaget razvil model, v katerem opisuje kako ljudje osmišljamo svet okoli sebe. Ta proces osmišljanja poteka z zbiranjem in organizacijo informacij, ki jih sprejemamo. Piagetov kognitivni model pojasnjuje in opisuje razvoj mišljenja od zgodnjega otroštva do odraslosti.

Od samega rojstva do odraslosti se procesi mišljenja počasi, a radikalno spreminjajo. Ključnega pomena pri kognitivnem razvoju je več kot le navajanje dejstev in idej v že obstoječe skladišče informacij. Na spremembe mišljenja vplivajo naslednji štirje dejavniki, ki jih je določil Piaget:

- Biološka maturacija

Gre za biološke spremembe, na katere sami, oziroma kot starši in učitelji, težko oziroma ne moremo vplivati. Pripomoremo lahko le toliko, da poskrbimo, da otroci dobijo dovolj potrebne nege in skrbi, ki jo potrebujejo za razvoj.

- Aktivnost

V času fizične maturacije otroci z delovanjem v okolju spreminjajo procese mišljenja preko raziskovanja, odkrivanja, preizkušanja, opazovanja in organiziranja informacij, ki jih od okolja sprejemajo. S fizično maturacijo se razvija tudi vedno večja sposobnost delovanja v okolju in otroci se soočijo z učenjem iz okolja.

- Socialne izkušnje

Poleg učenja od okolja v času fizične maturacije je pomembna tudi socialna transmisija. Otroci se soočijo z učenjem od drugih ljudi, s katerimi se srečujejo.

- Ekvilibracija

Ekvilibracija pomeni iskanje miselnega ravnotežja. Piaget je sklepal, da sta pomembnega pomena dve osnovni težnji: organizacija in adaptacija. Strukture, ki so naše orodje za razumevanje okolja in interakcijo z okoljem, je Piaget poimenoval sheme in le te so osnovni element mišljenja. Te sheme ves čas prilagajamo in svoje mišljenje organiziramo znotraj teh psiholoških struktur.

S prilagajanjem pa sta povezana še dva pomembna procesa. Proces asimilacije poskrbi, da za razumevanje nove situacije uporabimo že oblikovane sheme in poskušamo novo razumeti z že obstoječim znanjem. Proces akomodacije pa omogoča spreminjanje obstoječih shem, saj trenutne ne zadostujejo za razumevanje novih situacij.

### 2.1.1. Štiri faze kognitivnega razvoja

Piaget je kognitivni razvoj razdelil na štiri faze. Vsaka faza nastopi pri določeni starosti.

<b>FAZA</b>	<b>STAROST</b>	<b>ZNAČILNOST</b>
Senzomotorična	0-2 leti	Začne uporabljati imitacijo, spomin in mišljenje. Začne spoznavati, da objekti ne prenehajo obstajati, kadar jih skrijemo. Preide z refleksov na ciljno usmerjene aktivnosti.
Predoperacionalna	2-7 let	Postopoma začne uporabljati jezik in razvije sposobnost mišljenja v simbolni obliki. Sposoben je logično premisliti operacije v eno smer. Ima težave pri upoštevanju zornega kota druge osebe.
Konkretne operacije	7-11 let	Sposoben je rešiti konkretne (oprijemljive) probleme na logičen način. Razume zakone konzervacije in je sposoben klasifikacije in seriacije. Razume reverzibilnost.
Formalne operacije	11-odraslost	Sposoben je reševati abstraktne probleme na logičen način. Pri mišljenju postane bolj znanstven. Razvije skrb za socialne teme, identiteto.

Tabela 1: Faze kognitivnega razvoja po Piagetu (Woolfolk, 2002)

### ***2.1.2. Piagetov kognitivni razvoj in pouk***

Piaget pravi, da bomo uspešnejši pri poučevanju, če bomo razumeli kako otroci razmišljajo. S tem bomo namreč uspešnejši pri usklajevanju metod poučevanja s sposobnostmi otrok.

Najboljši vir informacij o učenčevih sposobnostih mišljenja so učenci sami (Confrey, 1990a).

Ne glede na to, v katerem razredu učenci so, se med seboj razlikujejo tako v akademskem kot kognitivnem razvoju. Težave lahko nastopijo ali zaradi pomanjkanja potrebnih sposobnosti mišljenja ali preprosto zato, ker se niso naučili osnovnih dejstev.

Za poučevanje je pomemben naslednji vidik Piagetove teorije, »problem ujemanja«. Tega je že pred leti tako imenoval Hunt (Hunt, 1961). Učenci se med poukom ne smejo dolgočasiti, a hkrati ne smejo zaostajati zaradi nerazumevanja snovi. Delo oziroma izziv mora biti učencem ravno pravšnji, torej ne pretežak in ne prelahak. Stopnja težavnosti mora biti ravno takšna, da v učencih spodbudi konflikt med tem, kar mislijo, da se bo zgodilo in med tem, kar se dejansko zgodi. Tako učenci reflektirajo situacijo, ponovno razmislijo o situaciji in razvije se lahko novo znanje (Woolfolk, 2002).

Bistvo Piagetove teorije je, da posamezniki konstruirajo svoje lastno znanje, saj je učenje konstruktiven proces. Da bi učenci informacije, ki jih prejemajo vključili v svoje sheme, morajo z njimi nekaj narediti. Vsak učenec potrebuje učitelje in vrstnike za interakcijo z njimi, saj tako testirajo svoje razmišljanje in doživljajo izzive, opazujejo kako drugi rešujejo probleme ipd. Ob interakciji z drugimi se disekvilibrirajo sproži sam od sebe.

### ***2.1.3. Omejitve Piagetove teorije***

Pojavilo se je kar nekaj kritik na Piagetovo teorijo kognitivnega razvoja. Marsikateri psihologi namreč niso verjeli v fazno razdelitev razvoja mišljenja. Problem naj bi bil v pomanjkanju konsistentnosti v otrokovem razmišljanju. Operacije, ki se jo je otrok naučil pri reševanju nekega problema, naj bi ponovno uporabil pri drugih problemih, ki zahtevajo to operacijo (Gelman & Baillargeon, 1983). Piaget naj bi tudi podcenjeval kognitivne sposobnosti otrok, predvsem najmlajših. Spregledal pa naj bi tudi pomembne vplive otrokove kulture in socialne skupine. Nekatere osnovne funkcije morda niso tako osnovne ljudem druge kulture.

Na podlagi teh in še nekaterih drugih kritik Piagetove teorije, je danes vplivnejši pogled na kognitivni razvoj, ki ga je predlagal Lev Vigotski – ta veže kognitivni razvoj na kulturo.

## 2.2. Konstruktivizem

Lev Semjonovič Vigotski, ruski psiholog, je verjel, da človeške aktivnosti ne moremo ločiti od okolja in jih razumeti ločene od okolja. Ključna ideja Vigotskega je, da specifične miselne strukture in procese zasledimo v interakcijah z drugimi. Socialne interakcije dejansko ustvarjajo kognitivne strukture in procese mišljenja in niso le preprosti vplivi na kognitivni razvoj (Palinscar, 1998).

Vigotski je bil mnenja, da na otrokov kognitivni razvoj vplivata dve vrsti interakcije z okoljem:

- Interakcija z ljudmi

Vsaka funkcija v otrokovem kulturnem razvoju se pojavi dvakrat. Najprej na socialnem nivoju in kasneje na individualnem nivoju. Višji miselni procesi, kot naprimer reševanje problemov, se pojavi najprej med ljudmi, otrok nato te procese internalizira in postanejo del njegovega kognitivnega razvoja.

Primer deklice, ki je izgubila igračo in prosi očeta za pomoč, dobro ponazarja ta pogled na otrokov kognitivni razvoj. Deklica pravi: »Ne morem se spomniti, kje sem nazadnje videla igračo.« Oče ji nato postavi serijo vprašanj: »Si jo imela v sobi? Zunaj? Pri sosedih?« Na vsako vprašanje deklice odgovori: »Ne.« Ko pa jo vpraša: »Si jo imela v avtu?« ona reče »Mislim, da.« in gre po igračo (Tharp & Gallimore, 1988).

Tako je deklica lahko internalizirala strategijo iskanja izgubljene igrače in bo lahko naslednjič samostojna pri reševanju takega problema. Tako se najprej pojavijo višje funkcije med učiteljem in otrokom, šele potem pa obstajajo znotraj otroka samega. Za Vigotskega je bila socialna interakcija več kot le vpliv, bila je izvor višjih miselnih procesov.

- Interakcija s sredstvi kulture

Vigotski je bil prepričan, da sredstva kulture igrajo zelo pomembno vlogo v kognitivnem razvoju. Med ta sredstva štejemo resnične pripomočke (pisalni stroj, ravnala, računalna, danes mobilni telefoni, računalniki) in simbolne sisteme (matematični sistemi, Braillova pisava, znakovni jezik, zemljevidi, znaki in simboli ter jezik).

Vigotski (1977: 20) navaja: »...nenehni razvoj človekovega vedenja temelji prvotno na izpopolnjevanju zunanjih znakov, zunanjih metod in tehnik, ki se izgrajujejo v določenem socialnem okolju pod pritiskom tehničnih in ekonomskih potreb. Pod njihovim pritiskom se spreminjajo tudi vsi naravni psihološki procesi človeka. Tisto pomembno pri teh procesih pa

je, da se izpopolnjujejo, prihajajo od zunaj in so odvisni od socialnega življenja skupine ali naroda, kateremu posameznik pripada.« (Žist & Oblak, 2004).

Vigotski je poudaril sredstva, s katerimi kultura podpira razvoj mišljenja. Odrasli s temi sredstvi poučujejo otroke preko vsakodnevnih aktivnosti, otroci pa jih internalizirajo. Ob tem razvijajo znanje, ideje, stališča in vrednote otrok skozi primerne ali »privzete za njihove« načine delovanja in razmišljanja, ki ga nudi njihova kultura in bolj sposobni člani njihove skupine (Kozulin & Presseisen, 1995).

Piaget in Vigotski sta poudarjala pomembnost socialnih interakcij, vendar vsak na svoj način. Piaget poudarja, da interakcije spodbujajo razvoj z ustvarjanjem kognitivnega konflikta, ta pa posledično motivira spremembo. Vigotski pa je bil mnenja, da kognitivni razvoj otroka spodbujajo osebe, ki so spretnejše ali prednjačijo v svojem mišljenju, torej učitelji ali starši (Moshman, 1997; Palinscar, 1998).

### **2.2.1. Teorija Vigotskega in pouk**

Obstajajo trije načini prenašanja sredstev kulture z enega posameznika na drugega:

- učenje s posnemanjem (ena oseba poskuša posnemati drugo),
- učenje z navodili (učenci internalizirajo navodila učitelja in jih uporabijo pri samoregulaciji) in
- sodelovalno učenje (skupina vrstnikov si prizadeva razumeti drug drugega in v tem procesu se pojavi učenje) (Tomasello, Kriger, & Ratner, 1993).

Eden glavnih vidikov poučevanja v vsaki situaciji je učenje s pomočjo.

### **2.2.2. Učenje s pomočjo**

Ni dovolj le urediti učnega okolja tako, da otroci samostojno odkrivajo, potrebno je več. Tako je videl Vigotski glavno vlogo učiteljev, staršev in drugih odraslih pri otrokovem učenju in razvoju.

Učenje s pomočjo ali vodeno sodelovanje temelji na podajanju informacij, pohval, spodbud, namigov, spodbujanja pri vztrajanju v pravem času in v pravi količini seveda. To imenujemo »zidarski oder« (ang. scaffolding). Kasneje pa dopuščamo učencem, da vedno več storijo sami. V kolikšni meri ponuditi pomoč in kakšno pomoč ponuditi pa nam delni odgovor da vedenje o območju bližnjega razvoja.

### **2.2.3. Območje bližnjega razvoja**

Tekom otrokovega razvoja se na katerikoli točki pojavljajo določeni problemi, ki jih je otrok skoraj že sposoben samostojno rešiti. Za uspeh pri reševanju pa pri tem potrebuje nekaj

namigov, nasvetov, pomoči pri pomnjenju korakov ali podrobnosti, nekaj ključnih točk in spodbudo pri vztrajanju.

Območje, kjer otrok ni sposoben samostojno rešiti problema sam, ampak s pomočjo spretnejšega oziroma sposobnejšega vrstnika, imenujemo območje bližnjega razvoja (Wertsch, 1991). V tem področju je lahko pouk uspešen, saj je predpogoj za učenje.

### **2.3. Razlika med konstruktivizmom in konstrukcionizmom**

Seymour Aubrey Papert je človek, delujoč na področjih matematike, poučevanja in računalništva. Na pariški univerzi je še kot študent spoznal Piageta, katerega delo je močno vplivalo nanj in je tako razvil vejo učnih teorij, imenovano konstrukcionizem.

Konstrukcionizem temelji na idejah konstruktivistične teorije. Z gradnjo miselnih modelov otroci skušajo osmisliti svet okoli sebe. Velja pa, da je učenje najbolj učinkovito, ko je otrok sam aktiven pri delu oziroma, ko povežemo aktivnosti z izkustvenim učenjem (Wikipedia, 2014).

Papert je zapisal:

*»Konstrukcionizem si s konstruktivizmom deli skupno idejo učenja, tj. gradnje znanja, ne glede na to, v kakšnih okoliščinah učenje poteka. Dodaja pa idejo, da posameznik zavestno sodeluje pri gradnji znanja. Ta proces poteka tesno povezano z okolico in neglede na kompleksnost znanja, ki ga gradi.«<sup>1</sup>*

Papert se je osredotočil bolj na učenje skozi »ustvarjanje« kot na sam kongitivni pristop učenja. S tem nam omogoča lažje razumevanje kako se ideje oblikujejo in preoblikujejo, ko jih prikažemo z različnimi oblikami predstavitev, ko jih udejanjimo v določenem kontekstu in ko jih izdelata vsak posameznik. Ključ do učenja, meni Papert, je projiciranje notranjih občutkov in idej.

Poudarjanje zunanjih vplivov na razvoj posameznikovega mišljenja ni nekaj novega. Vigotski je ves čas svojega življenja namenil raziskovanju, kako sredstva kulture (orodja, jezik, ljudje)

---

<sup>1</sup> Prevedeno in povzeto po članku Seymour Papert's Legacy: Thinking about learning, and learning about thinking avtorja Paulo Blikstein (Stanford University): Constructionism shares constructivism's connotation of learning as 'building knowledge structures' irrespective of the circumstances of the learning. It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe.



vplivajo na razvoj posameznikovega mišljenja in kognitivnega potenciala. Med konstruktivizmom in Papertovem konstrukcionizmom zasledimo tri bistvene razlike:

1. vloga zunanje pomoči vpliva na razvoj višjih miselnih procesov posameznika
2. pomembne so oblike zunanje pomoči (Papert se je bolj osredotočil na pomoč, ki temelji na uporabi računalniško podprte tehnologije)
3. vrste pobud/pohval, ki jih posameznik prejme pri oblikovanju sredstev, s katerimi osmišlja svet okoli sebe

Glavna Papertova ideja je bila naučiti otroke uporabe računalnikov. Ne le kot pasivne uporabnike ponujene opreme, temveč kot ustvarjalce nove računalniške opreme. Tako je na Tehnološkem inštitutu Massachusettsa sklical simpozij z naslovom »Teaching children thinking« oziroma »Naučiti otroke razmišljati«. Ne preseneča, da so pozneje ravno tam, nasledniki Papertove raziskovalne skupine ustvarili Scratch.

#### **2.4. Uporaba Scratcha in konstrukcionizem**

Virtualni svet današnjim generacijam ni tuj. Otroci obvladajo veščine kot so igranje računalniških iger, brskanje po internetu, pisanje sporočil prijateljem, vključevanje v socialna omrežja, itd. Le redko kdo pa zaide v smer kreativnega ustvarjanja (Resnick, 2012). Scratch je eno izmed orodij, ki to spodbuja in želi, da bi otroci spoznali in uporabljali tehnologijo z druge perspektive.

Uporaba Scratcha omogoča učenje naslednjih veščin:

- Kreativno mišljenje: otroci se z ustvarjanjem figur in zgodb v svojih Scratch projektih razvijajo kot kreativni ustvarjalci. Uporabljajo svojo domišljijo in raziskujejo nove ideje.
- Sistematičnost: gradnja blokov, kljub svoji enostavnosti, zahteva sistematičnost za pravilno in natančno izvajanje programa.
- Kolaborativnost: Scratch skupnost omogoča sodelovanje z drugimi v obliki komentiranja objavljenih projektov, izdelovanja skupnih projektov, ustvarjanja skupnih Scratch vodičev in drugih aktivnosti.

Z izdelovanjem Scratch projektov se naučijo tudi drugih matematičnih in računalniških konceptov kot so spremenljivke, pogoji, dogodki in paralelizem. Vendar ključno vlogo imajo kreativno mišljenje, sistematičnost in kolaboracija, ki niso pomembni zgolj za programerje.

Te sposobnosti nam omogočajo lažje delovanje v družbi, tako v profesionalnem življenju kot v osebnih odnosih (Resnick, 2012).

### 3. VIZUALNA PROGRAMSKA OKOLJA

Učenje programiranja zahteva veliko discipline in časa, da se pokažejo rezultati, ki so zanimivi in zabavni (Lajovic, 2011). Ker pa učenje programiranja v številnih pogledih dobro koristi razvijanju mišljenja in reševanju problemov ter nudi usvajanje osnovnih življenjskih veščin današnjega digitalnega sveta, je koristno s tovrstnim učenjem začeti že v mlajših letih. Da pa bi otrokom svet programiranja bil zabaven in zanimiv, je potrebno prestrukturirati način pisanja kode. Tu pridejo na vrsto vizualna programska okolja, ki zahtevno pisanje kode in dolgotrajno ter zahtevno programiranje spremeni v svet zabavnega učenja.

Vizualna programska okolja namesto pisanja kode za sestavo programa uporabljajo grafične elemente, kot so simboli ali ikone. Uporabnik pri pisanju programa manipulira z grafičnimi elementi in jih »sestavlja« v neko zaporedje, ki služi gradnji programa (Wikipedia, Visual programming language. The Free Encyclopedia, 2014).

Vizualna programska orodja za poučevanje programiranja mlajših ima dve glavni prednosti. Prva je koncept WYSIWYG (what you see is what you get – to kar vidiš to, dobiš/narediš), saj nudi takojšnjo povratno informacijo. Druga pa je odprava sintaktičnih napak, saj do njih praktično ne more priti, saj znanje sintakse v tem primeru ni potrebno (Constructing Kids, 2013).

Vizualnih programskih okolij, ki so namenjena učenju programiranja osnovnošolskih otrok, je danes na voljo kar nekaj. V nadaljevanju je na kratko opisanih nekaj takšnih vizualnih programskih okolij, ki so primerna za poučevanje v osnovni šoli.

#### 3.1. Scratch

Scratch je eden novejših vizualnih programskih orodij, ki je namenjen zgolj poučevanju programiranja za otroke od 8. leta starosti naprej. Otroci lahko brez predznanja ustvarjajo svoje lastne animacije, računalniške igre in interaktivne zgodbe ter jih tudi delijo z ostalimi po svetu, kar omogoča dodatno izpopolnjevanje in izmenjavo izkušenj (Lajovic, 2011).

Scratch je brezplačno orodje in omogoča, da se otroci že v



Slika 1: Scratch maček

mladih letih naučijo kreativnega razmišljanja in ustvarjanja, kar je v današnjem času izredno pomembno. S takšnim ustvarjalnim delom omogočajo otrokom, da se naučijo osnovnih življenjskih veščin današnjega časa (MIT, Scratch - Zamisli si, programiraj, deli, 2014).

Otrok lahko s pomočjo Scratcha spozna naslednje matematične in računalniške koncepte: prireditveni stavek (spremenljivka), zanka, pogojni stavek, tabela, odziv na dogodek, časovna usklajenost in sinhronizacija, vnos podatkov preko tipkovnice, naključna števila, logične operacije in dinamična interakcija (Lajovic, 2011), podprograme in rekurzije (MIT, Scratch - Zamisli si, programiraj, deli, 2014). Poleg zgoraj naštetih matematičnih in računalniških konceptov pa lahko otrok v program vključi tudi koncepte večpredstavnosti (glasbo, grafiko, ...) in spozna metode načrtovanja programa. Torej lahko rečemo, da Scratch poleg tega, da otroka uči sistematičnega in strukturiranega načina razmišljanja ter ga skozi igro vodi do cilja, spodbuja tudi kreativnost, jasno komunikacijo, uporabo sodobnih tehnologij, interaktivno načrtovanje in stalno učenje (Stopar et al., 2014).

Program so razvili na inštitutu Media Lab pod vodstvom Mitchela Resnicka. Sestavljanje programa pa otrokom olajšujejo bloki v različnih barvah, ki predstavljajo raznovrstne ukaze, samo sestavljanje pa nudi dodatno zabavo pri delu (Lajovic, 2011), (MIT, Scratch - Zamisli si, programiraj, deli, 2014).

ScratchEd je spletna skupnost, ki je namenjena predvsem tistim, ki poučujejo program Scratch za izmenjavo virov, povezovanje z drugimi, postavljanje morebitnih vprašanj in deljenje zgodb (About. ScratchEd, 2014).

Program Scratch je že prešel iz prve večje verzije na drugo, ki je izšla januarja 2013. Scratch 2.0 je tako druga večja verzija programa, ki ima kar nekaj novosti, kar omogoča toliko več zabave pri programiranju (Scratch 2.0 - Scratch Wiki, 2014).

Pred kratkim, od avgusta 2014, pa je na voljo tudi aplikacija ScratchJr, ki jo zaenkrat lahko namestimo na tablice iPad. Namenjena je otrokom starim med 5 in 7 let in omogoča ustvarjanje interaktivnih zgodbic in iger. Otroci z zlaganjem blokov lahko pripravijo svojo figuro do premikanja, skakanja, plesanja in petja. Ustvarijo lahko tudi svoje lastne ilustracije, posnamejo svoj glas in vključijo zvok ali slike (MIT, ScratchJr - About, 2014).

ScratchJr ima popolnoma prenovljen vmesnik in programski jezik, saj so ga želeli čimbolj prilagoditi starostni skupini otrok med 5 in 7 letom starosti tako, da je čimbolj v skladu z njihovim kognitivnim, socialnim, osebnim in čustvenim razvojem (MIT, ScratchJr - About, 2014).

V koncu leta 2014 se načrtuje razvoj aplikacije za operacijski sistem Android in leta 2015 še razvoj spletne verzije aplikacije (MIT, ScratchJr - About, 2014).



Slika 2: ScratchJr

### 3.2. Alice 3D

Alice 3D je inovativno programsko okolje, ki otroke uči računalniškega programiranja v 3D okolju. Je prosto programsko orodje, ki omogoča enostavno kreiranje animacij za pripovedovanje zgodb, igranje interaktivnih iger in ustvarjanja videa in tako otrokom omogoči enostavno prvo srečanje z objektnim programiranjem, kot je npr. programiranje v programskem jeziku Java. Uporabniški vmesnik je za otroke nezahteven, saj sestavljanje programa omogoča z ukazom »povleci in spusti« in ponuja takojšnjo povratno informacijo (Alice.org, 2014).



Slika 3: Alice logotip

Programsko okolje Alice 3D je razvila raziskovalna ekipa na univerzi University of Virginia in Carnegie Mellon leta 1998 pod vodstvom Randy Pausch (Alice.org, 2014).

### 3.3. Kodu

Kodu je novejšo vizualno programsko orodje v prvi vrsti namenjeno kreiranju iger. Nastal je pod okriljem Microsoft podjetja leta 2009, uporaben pa je tako za najmlajše kot tudi starejše, saj znanje programiranja ni pogoj za uporabo tega programskega okolja. Kodu spodbuja kreativnost in poleg učenja programiranja omogoča učenje spretnosti pri reševanju problemov in pripovedovanja zgodb (Kodu - Microsoft Research, 2014).



Slika 4: Kodu logotip



Slika 5: Uporabniški vmesnik programa Kodu

Programski jezik, ki ga Kodu uporablja, je enostaven in v celoti temelji na ikonah, ki jih je potrebno pravilno »zložiti« v zaporedje. Programi, ki jih je mogoče ustvariti, so sestavljeni iz strani, te pa so razčlenjene na pravila, ki se ločijo na pogoje (angl. When) in akcije (angl. Do). Vsak stavek znotraj programa, ki ga sestavimo, temelji na principu: »*When something happens, do something*« (Kodu - Microsoft Research, 2014). Programski jezik je posebej zasnovan za razvoj iger in vsebuje okolja, ki izhajajo iz igralnih scenarijev (Glastovec, 2012).

Na domači spletni strani programa Kodu najdemo forum, ki je namenjen tako izmenjavi izdelkov kot postavljanju vprašanj ter objavljanju novih informacij. Namenjen pa je tudi učiteljem in poučevanju programiranja s pomočjo Kodu programa. Preko foruma je učiteljem omogočena medsebojna pomoč pri poučevanju s programom Kodu in izmenjavanju idej ter informacij (Kodu - Microsoft Research, 2014).

Žal Kodu skupnost, ki je bila namenjena ustvarjalcem in učiteljem programiranja s programom Kodu in je bila na naslovu [www.planetkodu.com](http://www.planetkodu.com), ne deluje več (Kodu - Microsoft Research, 2014).

## 4. PROGRAMIRANJE V OSNOVNI ŠOLI

V poglavju bomo opisali učni načrt novega neobveznega izbirnega predmeta Računalništvo in ga primerjali z učnim načrtom že obstoječih predmetov, ki v šolah poučujejo računalniške vsebine. Poglavje je v celoti povzeto po učnih načrtih, ki ju je izdalo Ministrstvo za šolstvo, znanost in šport.

Danes je v šoli računalništvo prisotno v obliki učenja veščin uporabe računalnika in ne kot razumevanje osnovnih konceptov računalništva. Trenutno se vsebine iz področja računalništva poučujejo pri predmetu Računalništvo, ki je umeščen v tretje vzgojno-izobraževalno obdobje osnovne šole, 7., 8. in 9. razred. Pokriva pa le naslednje vsebine:

- Multimedija,
- Urejanje besedila in
- Računalniška omrežja (Ministrstvo za šolstvo, 2002).

S temi kurikularnimi vsebinami, ki jih zajemajo zgornji predmeti, otroke učimo veščin uporabe računalniške programske opreme, s katerimi pridobijo del računalniške pismenosti, ki je potrebna pri nadaljnjem izobraževanju in vsakdanjem življenju. Vendar pa takšno znanje »zastara« v hitrem in nenehnem razvoju tehnologije.

Z uvajanjem novega neobveznega izbirnega predmeta Računalništvo pa bomo otroke lahko naučili razumevanja osnovnih konceptov računalništva. Predmet Računalništvo se uvaja v drugo vzgojno-izobraževalno obdobje, v 4., 5. in 6. razred, zajema pa predvsem vsebine programiranja in je razdeljen na področja Algoritmi, Programi, Podatki, Reševanje problemov in Komunikacija in storitve. Način dela pri predmetu spodbuja ustvarjalnost, sodelovanje in poseben način razmišljanja in delovanja. S spoznavanjem računalniških konceptov in razvijanjem postopkovnega načina razmišljanja učenci pridobivajo znanja, spretnosti in veščine, ki so veliko bolj trajni kot hitro razvijajoče se tehnologije.

Spodnja primerjava splošnih učnih ciljev nam da vpogled v raznolikost vsebin poučevanja in vpogled v vsebinsko usmerjenost novega predmeta.

<b>Splošni cilji predmeta</b>	
<b>Urejanje besedila, Multimedija in Računalniška omrežja</b>	<b>Računalništvo</b>
<p>Učenci in učenke:</p> <ul style="list-style-type: none"> <li>• spoznavajo osnovne pojme računalništva in vlogo ter pomen računalniške tehnologije v sodobni družbi,</li> <li>• spremljajo razvoj računalniške tehnologije,</li> <li>• pridobivajo temeljna znanja, spretnosti in navade za učinkovito ter uspešno uporabo sodobne računalniške tehnologije za zadovoljevanje svojih in družbenih potreb,</li> <li>• razvijajo komunikacijske zmožnosti,</li> <li>• oblikujejo stališča do pridobljenih informacij in krepijo merila za doživljanje ter vrednotenje lepega,</li> <li>• razvijajo sposobnosti za učinkovito in estetsko oblikovanje informacij,</li> <li>• pridobivajo sposobnost samostojnega reševanja problemov,</li> <li>• razvijajo sposobnost in odgovornost za sodelovanje v skupini ter si krepijo pozitivno samopodobo,</li> <li>• razvijajo pravilen odnos do varovanja lastnine (avtorske pravice) in osebnosti (zaščita podatkov) ter</li> <li>• bogatijo svoj jezikovni zaklad in skrbijo za pravilno slovensko izražanje.</li> </ul>	<p>Pri predmetu učenci:</p> <ul style="list-style-type: none"> <li>• spoznavajo temeljne koncepte računalništva,</li> <li>• razvijajo algoritmični način razmišljanje in spoznavajo strategije reševanja problemov,</li> <li>• razvijajo sposobnost in odgovornost za sodelovanje v skupini ter si krepijo pozitivno samopodobo,</li> <li>• pridobivajo sposobnost izbiranja najustreznejše poti za rešitev problema,</li> <li>• spoznavajo omejitve človeških sposobnosti in umetne inteligence,</li> <li>• se zavedajo omejitev računalniških tehnologij,</li> <li>• pridobivajo zmožnost razdelitve problema na manjše probleme,</li> <li>• se seznanjajo z abstrakcijo oz. poenostavljanjem,</li> <li>• spoznavajo in razvijajo zmožnost modeliranja,</li> <li>• razvijajo ustvarjalnost, natančnost in logično razmišljanje ter</li> <li>• razvijajo in bogatijo svoj jezikovni zaklad ter skrbijo za pravilno slovensko izražanje in strokovno terminologijo.</li> </ul>

Tabela 1: Primerjava učnih ciljev (Ministrstvo za šolstvo, 2002), (Ministrstvo za šolstvo, 2013)

Ključnega pomena pri poučevanju tega predmeta je ustvarjanje spodbudnega učnega okolja in okoliščin, ki omogočajo učencem odkrivanje, izgrajevanje in oblikovanje (spo)znanj ter razvijanje kritičnosti in odgovornosti. Za doseg tega pa je nujno, da pouk temelji na učenčevi aktivni usvojitvi znanja na način, da so v ospredju dejavnosti učencev, ki to omogočajo. Učiteljeva vloga je ustvarjanje okoliščin, ki bodo takšno obliko poučevanja spodbujale.

Učni načrt omogoča tudi več možnosti medpredmetnega povezovanja, ne le vsebinskega, saj omogoča razvijanje spretnosti, ki so uporabne v različnih okoliščinah (npr. kritično mišljenje, obdelava podatkov, uporaba informacijsko – komunikacijske tehnologije). Za učitelje izvajalce so tovrstna znanja pogoj za kvalitetno poučevanje izbirnih predmetov s področja računalništva in informatike.

Med vsemi vizualnimi programskimi okolji smo izbrali Scratch, saj je najbolj skladen z učnim načrtom predmeta Računalništvo in v je skladu z ugotovitvami iz strokovne literature primeren za uvod v programiranje.



## 5. EMPIRIČNA RAZISKAVA

Namen empirične raziskave je bilo približno oceniti mnenje učiteljev o lastni pripravljenosti na izvajanje novega neobveznega izbirnega predmeta Računalništvo, s poudarkom na poučevanju programiranja. Še pred oblikovanjem učnih gradiv nas je zanimal odnos učiteljev do vsebin v novem učnem načrtu. Zanimalo nas je, za katere vsebine potrebujejo pomoč v obliki učnih gradiv.

Empirično raziskavo smo izvedli s kvalitativno metodo raziskovanja in sicer kratek intervju, ki je vseboval osem vprašanj. Sodelovalo je pet osnovnošolskih učiteljev.

Z izvedbo intervjuja nas je zanimalo tudi:

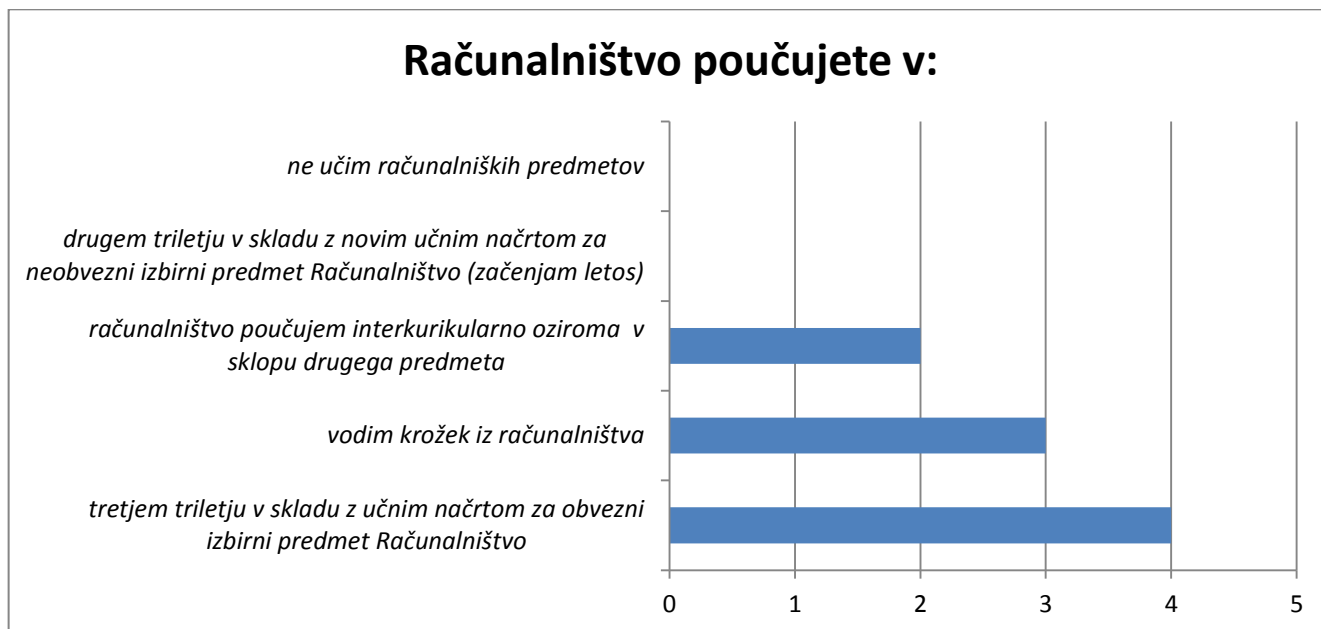
- če se učitelji dodatno izobražujejo,
- na kakšen način se dodatno izobražujejo in
- kakšne možnosti bi si še želeli poleg objavljanja in izmenjavanja gradiv na spletnem portalu, kjer bodo le ta objavljena.

## Analiza odgovorov

### 1. Na kateri šoli poučujete?

Vsi intervjuvanci poučujejo na osnovni šoli.

### 2. Računalništvo poučujete v:



Graf 1: Oblika poučevanja računalništva

Vsi intervjuvanci poučujejo računalniške predmete. Štirje učitelji poučujejo predmet Računalništvo v tretjem triletju, v skladu z učnim načrtom za obvezni izbirni predmet Računalništvo, dva od njih pa računalništvo vodita tudi krožek iz računalništva, eden pa poučuje računalništvo interkurikularno oziroma v sklopu drugega predmeta.

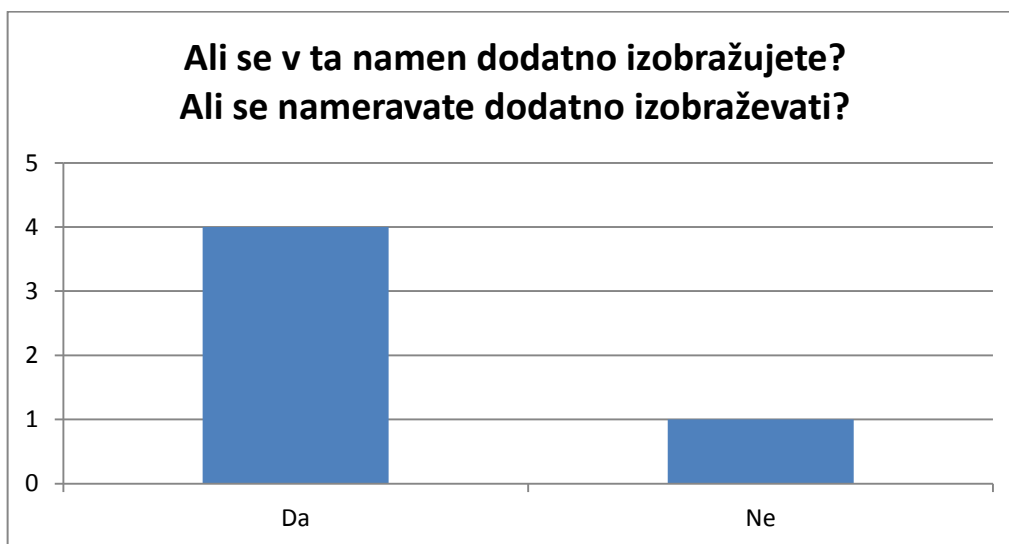
### 3. Koliko časa že poučujete računalniške predmete? (leta)

Vsi od intervjuvancev poučujejo računalniške predmete več kot 5 let. Eden jih poučuje 3 leta, trije poučujejo 5 let in eden poučuje 8 let.

4. V moji šoli se bo letos začel izvajati nov neobvezni izbirni predmet Računalništvo
- Da
  - Ne

Izmed intervjuvanih učiteljev se bo v letošnjem šolskem letu le en srečal z neobveznim izbirnim predmetom Računalništvo.

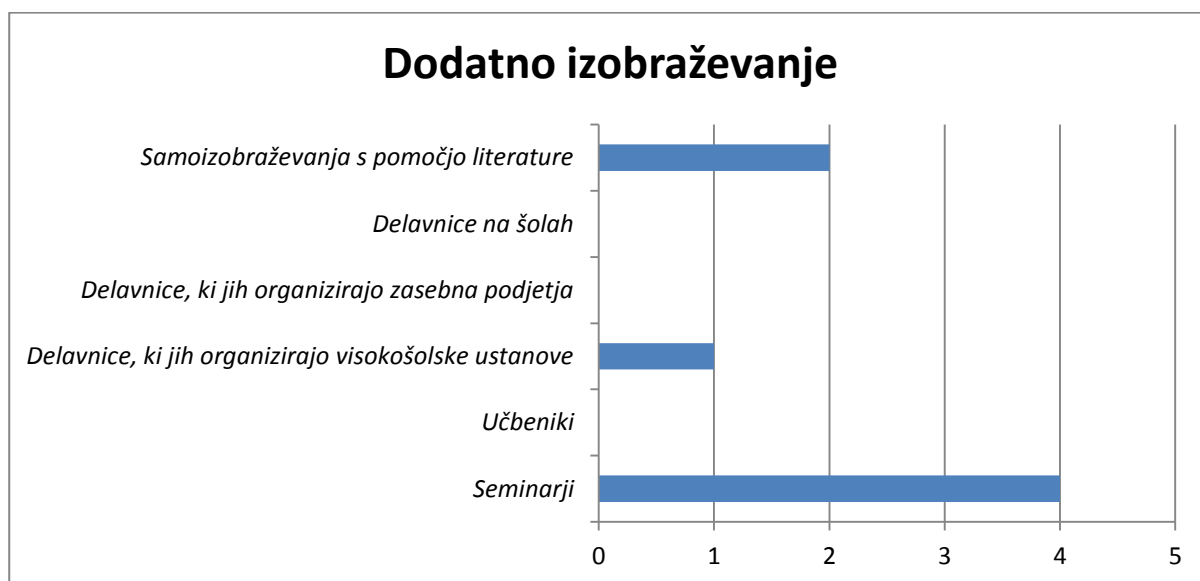
5. V osnovne šole se vpeljuje nov neobvezni izbirni predmet Računalništvo, ki bo vključeval računalniške vsebine, ki se do sedaj še niso poučevale.
- Ali se v ta namen dodatno izobražujete?
    - Da
    - Ne
  - Ali se nameravate dodatno izobraževati?
    - Da
    - Ne



Graf 2: Dodatno izobraževanje učiteljev

Zgornji dve vprašanji, 5a in 5b, sta imeli enake odgovore. Štirje učitelji se izobražujejo oziroma se imajo namen dodatno izobraževati, eden od učiteljev pa ne. Učitelji so tudi odgovarjali skladno, tisti, ki je pri vprašanju 5a odgovoril z da, je tudi z da odgovoril pri 5b.

c. V kakšni obliki potekajo izobraževanja?



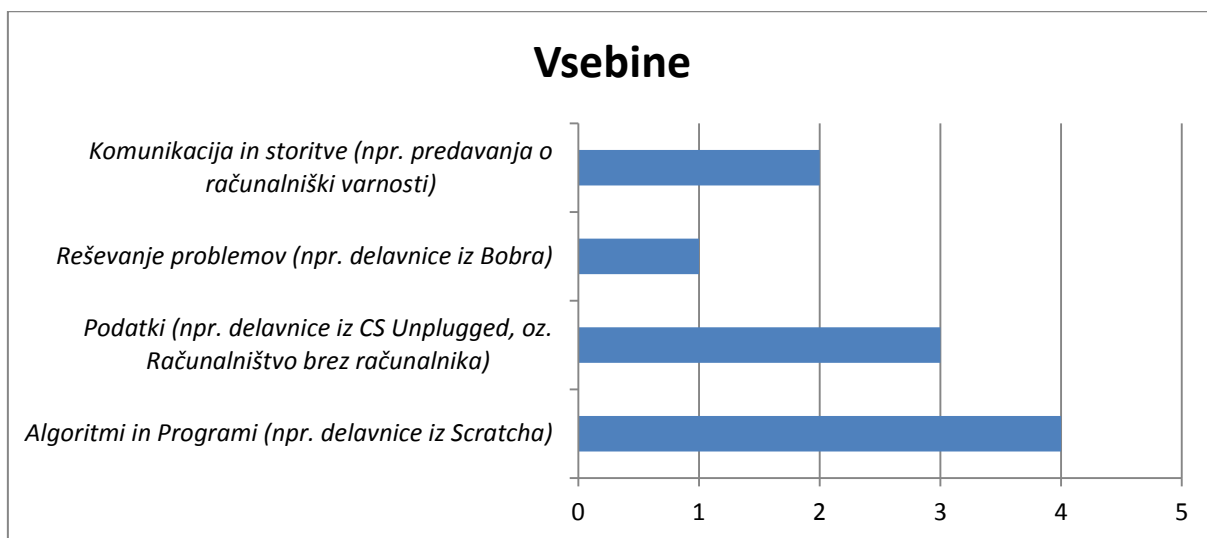
Graf 3: Oblike dodatnega izobraževanja učiteljev

Pri intervjuvanih učiteljih dodatna izobraževanja potekajo predvsem v obliki seminarjev in nekaj malega kot delavnice, ki jih organizirajo visokošolske ustanove ter v obliki samoizobraževanja.

Štirje učitelji se dodatno izobražujejo preko seminarjev, od tega eden obiskuje delavnice, ki jih organizirajo visokošolske ustanove in dva se dodatno izobražujeta samostojno s pomočjo literature.

Učitelj, ki se v letošnjem šolskem letu ne bo srečal z novim predmetov Računalništvo in se v ta namen tudi ne izobražuje dodatno, je pri tem vprašanju pod možnost *drugo* zapisal, da poučuje na mednarodnem oddelku, kjer je računalništvo vključeno v druge predmete in nimajo posebnega izobraževanja.

6. Za katere izmed učnih vsebin menite, da bi vam predstavljale večji izziv pri poučevanju novega neobveznega izbirnega predmeta Računalništvo? Za katere vsebine bi želeli imeti (dodatna) gradiva za izvajanje učnih ur?



Graf 4: Vsebine izbirnega predmeta Računalništvo

Največji izziv naj bi učiteljem predstavljala vsebina Algoritmi in Podatki, saj so to v ospredje postavili štiri učitelji. Od tega sta dva učitelja dodatno izpostavila še vsebino Podatki, eden vsebino Komunikacija in storitve, eden pa je izpostavil še vse preostale vsebine, torej kar vse iz učnega načrta. Eden izmed anketiranih učiteljev pa je izpostavil le vsebino Komunikacija in storitve.

7. V kolikšni meri se vam zdi pomembno in koristno izmenjati gradiva na spletu z ostalimi učitelji in si s tem pomagati pri izvajanju novega neobveznega izbirnega predmeta Računalništvo?

Ocena: 1 – najmanj pomembno, 4 – zelo pomembno

Pri tem vprašanju so se prav vsi intervjuvanci strinjali, da je zelo pomembno in tudi koristno izmenjavati gradiva z ostalimi učitelji in si tako pomagati pri poučevanju novega neobveznega izbirnega predmeta Računalništvo.

8. *Kakšne možnosti bi si na portalu, kjer bodo objavljena gradiva, še želeli poleg izmenjevanja (nalaganje gradiv na portal, prenos gradiv na računalnik)?*

Pri možnostih na portalu so učitelji imeli dokaj podobne ideje oziroma želje. Možnost komentiranja učnih gradiv oziroma postavitve foruma, znotraj katerega bi si lahko izmenjavali informacije in komentirali učna gradiva, je bila želja vseh intervjuvancev. Izpostavljena pa je bila tudi ideja o objavljanju novejših verzij učnih priprav, v smislu možnih izboljšav izpeljave učne ure.

### **Ugotovitve**

Iz odgovorov intervjuvancev smo lahko zaključili, da se učitelji pripravljajo na uvajanje novega neobveznega izbirnega predmeta Računalništvo, kljub temu, da se bo v letošnjem letu s tem predmetom srečal le eden izmed intervjuvanih učiteljev. Izredno pozitivne odgovore smo dobili o postavitvi portala, saj se jim pomoč drugih učiteljev zdi primerna in zelo dobrodošla. Ne glede na leta poučevanja računalniških predmetov, večina učiteljev največji izziv pričakuje pri vsebini Algoritmi in Programi, kar lahko dodatno potrди raznolikost med starim in novim učnim načrtom računalniških predmetov.

Glede na pripravljanje učiteljev na nov računalniški predmet in njihovo željo po medsebojni pomoči lahko sklepamo, da imajo pred seboj določen izziv, za katerega si želijo dodatne pomoči.

## **6. PRIPRAVA GRADIV**

Za namene poučevanja novega predmeta Računalništvo smo opisali realizacijo ene izmed tem in postopek izdelave spletnega portala *RIS* ter opis možnosti, ki jih ponuja.

### **6.1. Pedagoška osnova**

Delo v programu Scratch ponuja učiteljem raznoliko poučevanje programerskih vsebin ter vsebin s področja razvoja in uporabe večpredstavnosti. V diplomskem delu smo se odločili za vsebine: *spoznavanje okolja Scratch, risanje po navodilih, zaznavanje v Scratchu na primeru labirinta in učenje programiranja preko zgodb in ugank*. Opisali bomo realizacijo prve teme *Spoznavanje okolja Scratch*, ostale bomo le opisali.

#### **6.1.1. Risanje po navodilih**

Pri risanju po navodilih učenci utrdijo spoznanje, da računalniki med izvajanjem programa izvajajo zaporedje navodil. Če smo pri podajanju navodil – se pravi programiranju – nepazljivi, so lahko rezultati napačni ali vsaj nenatančni. Iz Scratcha spoznajo paletu Svinčnik.

S pomočjo figure najprej rišejo enostavne like (pravokotnik, krožnica, ipd.), pozneje pa tudi bolj kompleksne, kot je npr. Kohova snežinka.

#### **6.1.2. Zaznavanje v Scratchu na primeru labirinta**

Pri izdelavi labirinta se učenci srečajo predvsem z ukazi iz skupine Zaznavanje. Pri labirintu je pomembno, da se figura ne zaleti v stene, se izogiba oviram in drugim preprekam, ki so na poti do cilja. Na cilju pa je potrebno zaznati konec igre. Spoznajo se tudi z ukazi Premikanja, saj morajo določiti tipke za premikanje figure po labirintu. Poleg tega se učenci srečajo tudi z risalno površino, kjer lahko narišejo svoj labirint ali svojo figuro, ter z dodajanjem zvoka ob dosegu cilja in ostalimi ukazi, ki popestrijo igro.

#### **6.1.3. Učenje programiranja preko zgodb in ugank**

Pri učenju programiranja je bistveno, da se učencem predstavijo pojme kot so zanka, pogoj in spremenljivka. Te osnovne pojme z lahkoto zamaskiramo v zgodbe in uganke.

#### 6.1.4. Spoznavanje okolja Scratch

##### Opis

V učni uri *Spoznavanje okolja Scratch* učence spoznamo z vizualnim okoljem Scratch. Spoznajo način dela v Scratchu in osnovne funkcije programa. Predstavimo jim že objavljene projekte na Scratch domači spletni strani, s katerimi pokažemo različne možnosti ustvarjanja z omejenim naborom ukazov. Učencem med uro ponudimo Scratch karte, ki so jim v oporo pri raziskovanju in učni list z enostavnimi nalogami. Ob koncu ure se predstavi še projektna naloga *O meni*, ki jo mora opraviti vsak učenec posebej.

##### Realizacija teme

<b>Šola:</b> OŠ
<b>Razred:</b> 4
<b>Predmet:</b> Računalništvo
<b>Učna tema:</b> Algoritmi (in Programi)
<b>Učne oblike:</b> frontalna, delo v paru/skupini, individualno delo
<b>Učne metode:</b> metoda pogovora in razgovora, metoda demonstracije, metoda razlage, diskusije
<b>Namen</b> Učenci <ul style="list-style-type: none"><li>▪ spoznajo osnovne gradnike vizualnega programskega orodja Scratch,</li><li>▪ spoznajo, da lahko v vizualnem programskem okolju Scratch ustvarjajo like, okolja, zgodbe, animacije in igre,</li><li>▪ spoznajo, kaj je ukaz in</li><li>▪ znajo rešitev vsakdanjega problema opisati kot zaporedje korakov.</li></ul>
<b>Pripomočki:</b> <ul style="list-style-type: none"><li>• Scratch karte</li><li>• Datoteka <i>Enostavne naloge v Scratchu</i></li></ul>



## Uvodni del

ČAS (min)	UČITELJ	UČNE OBLIKE, METODE, PRIPOMOČKI
8	<ul style="list-style-type: none"> <li>učencem predstavimo vizualno programsko okolje Scratch z obiskom spletne strani. Skupaj z učenci si ogledamo nekaj že objavljenih projektov in s tem prikažemo različne možnosti ustvarjanja (zgodbe, animacije in igre). (npr.: Klavir, Najstniki v gradu in Odbij žogico)</li> <li>učencem predstavimo okolje, znotraj katerega nastajajo projekti (oder, ukazna paleta, programsko okno, seznam figur, risalna površina,...)</li> </ul>	<p>Frontalna, metoda pogovora in razgovora, metoda diskusije, domača stran Scratch <a href="http://scratch.mit.edu">http://scratch.mit.edu</a> projekti, objavljeni na strani <a href="http://scratch.mit.edu/starter_projects/">http://scratch.mit.edu/starter_projects/</a></p>

## Glavni del

ČAS (min)	UČITELJ	UČNE OBLIKE, METODE, PRIPOMOČKI
15	<p>Skupaj z učenci se pripravimo na delo v programu Scratch. Predstavimo jim ukaze, ki so razdeljeni v 8 skupin in pojasnimo, da jih v programskem oknu zlagamo skupaj kot LEGO kocke.</p> <p>Za demonstracijo mački sestavimo takšne ukaze, da bo mačka zaplesala. Ob tem razložimo naslednje skupine ukazov: <i>Premikanje</i>, <i>Zvok</i>, <i>Dogodki</i> in <i>Krmiljenje</i>.</p> <p>Najprej iz skupine <i>Premikanje</i> vzamemo ukaz »pojdi 10 korakov«. Tako pripravimo mačko, da se premakne vsakič, ko kliknemo na blok. Lahko</p>	<p>Frontalna, metoda demonstracije, individualno delo</p>

	<p>spremenimo število korakov, da se mačka premakne za večjo ali manjšo razdaljo.</p> <p>Nato iz skupine <i>Zvok</i> dodamo ukaz »zaigraj na boben 1 za 0,25 udarcev« in ga pripravimo pod ukaz »pojdi 10 korakov«. Pritisnemo na sestavljen blok in preizkusimo.</p> <p>Nato podvojimo nastali blok in ga pripravimo pod obstoječi blok. Pri tem v ukazu »pojdi 10 korakov«, spremenimo število korakov na -10, da se mačka premakne v nasprotno smer kot na začetku.</p> <p>Iz skupine <i>Krmiljenje</i> vzamemo ukaz »ponovi 10 - krat«, s katerim zaobjamemo že obstoječi blok.</p> <p>Manjka še ukaz »ko je ta figura kliknjena« iz skupine <i>Dogodki</i>. Postavimo ga na vrh že sestavljenih ukazov.</p> <p>Sedaj z miško pritisnemo na mačko in ta zapleše.</p> <p>S spreminjanjem vrednosti znotraj ukazov pokažemo, da lahko spreminjamo razdaljo mačkinega sprehoda, zvok in hitrost plesa.</p> <p>Učenci imajo na voljo tri minute za preizkus.</p> <p>(Dodatno: Plešočo mačko postavimo na oder. Učencem lahko pokažemo spreminjanje ozadja. S klikom na ozadje v seznamu figur izberemo novo ozadje iz knjižnice: <i>Spotlight-stage</i>. Kliknemo na mačko in ta zopet zapleše, tokrat na odru. )</p>	
25	<p>V nadaljevanju ure spodbudimo samostojno delo učencev.</p> <p>Učencem ponudimo Scratch karte, ki jih preizkusijo. Vsak učenec si izbere tri in jih samostojno preizkusi. Seveda je dovoljena medsebojna pomoč.</p> <p>(Dodatno: po preizkusu vseh treh kart lahko učenci poskusijo združiti poljubni dve karti ali pa</p>	Individualno delo, delo v paru, Scratch karte

	kar vse tri. Npr. učenec združi ukaze iz kart ( <i>Spremeni barvo in Premakni se v ritmu</i> )	
30	Za kartami sledi samostojno kreiranje programov. Vsakemu učencu podamo delovni list <i>Enostavne naloge v Scratchu</i> . Samostojno rešujejo zastavljene naloge. Pri delu si lahko pomagajo s Scratch kartami.	Individualno delo, Scratch karte, delovni list <i>Enostavne naloge v Scratchu</i>

### Zaključni del

ČAS	UČITELJ	UČNE OBLIKE, METODE, PRIPOMOČKI
12	<p>Za zaključek sledijo navodilom projektne naloge <i>Omeni</i>, ki jo vsak učenec izdelava samostojno.</p> <p>Vsak predstavi vsaj štiri lastnosti o sebi, ki jih želi deliti z ostalimi. Pri tem uporabi znanje, ki ga je do sedaj usvojil.</p> <p>Predstavitev projektnih nalog je uvod v naslednjo šolsko uro pred nadaljevalnim delom v Scratchu.</p>	Frontalna, metoda pogovora in razgovora, metoda diskusije

### Viri:

MIT. (2014). *Scratch - Zamisli si, programiraj, deli*. Pridobljeno iz <http://scratch.mit.edu/>

Lajovic, S. (2011). *Scratch. Nauči se programirati in postani računalniški maček*. Ljubljana: Pasadena.

MIT. (2011). Pridobljeno iz

<http://scratched.media.mit.edu/sites/default/files/CurriculumGuide-v20110923.pdf>

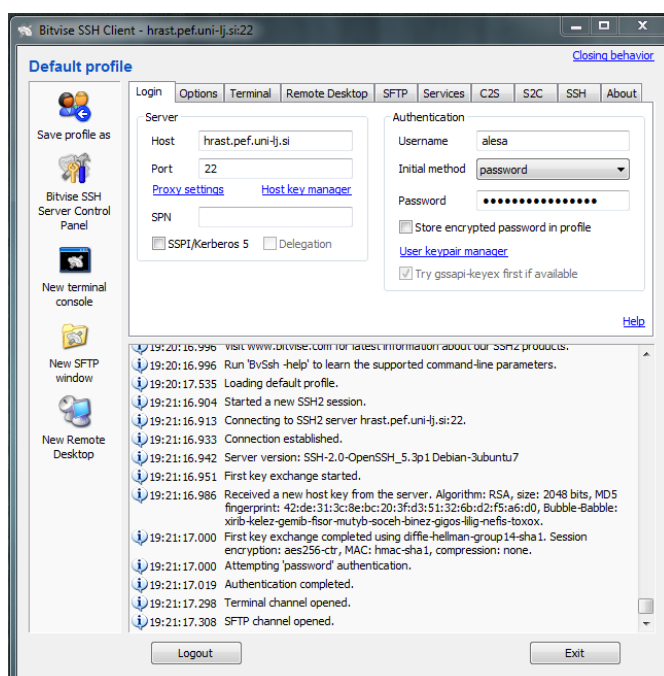
## 6.2. Tehnološko okolje – portal z gradivi

Gradiva, ki smo jih izdelali v diplomskem delu, so dostopna na spletnem portalu z imenom *RIS*, ki se nahaja na fakultetnem strežniku *Hrast* (*hrast.pef.uni-lj.si /ris*). Spletni portal *RIS* je nastal v okolju *Wordpress* in je namenjen računalniškemu izobraževanju s *Scratchem*.

Na portalu so objavljena kvalitetna učna gradiva, kot pomoč učiteljem pri poučevanju predmeta Računalništvo in so dostopna vsem. Vsako gradivo je mogoče prenesti na računalnik in tudi komentirati, s čimer želimo spodbuditi medsebojno pomoč učiteljev.

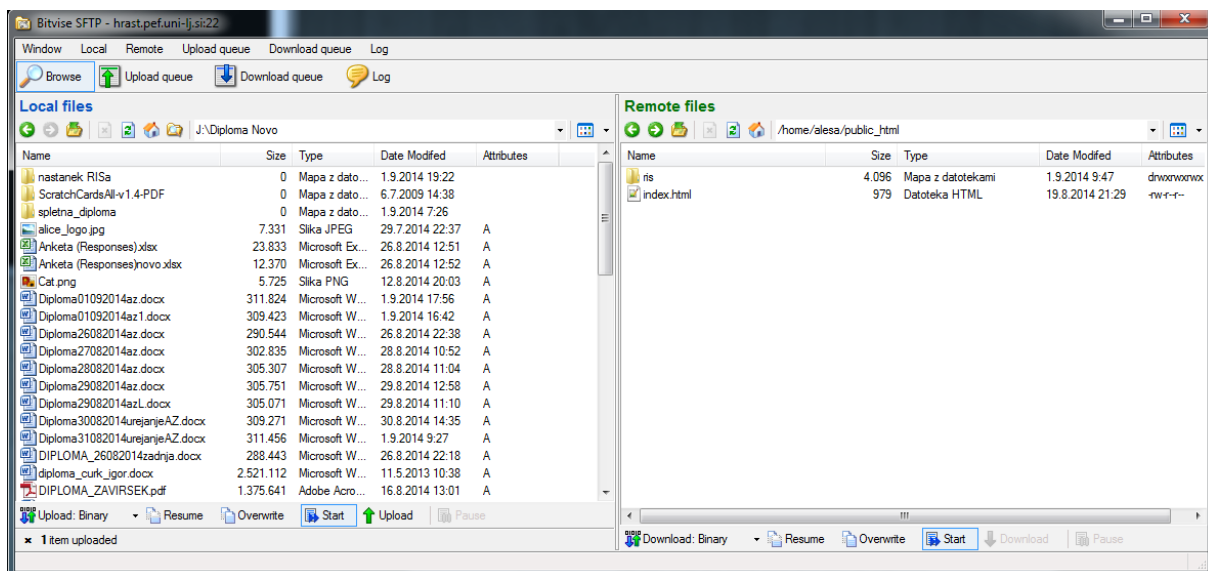
### 6.2.1. Kratek opis postavitve spletnega mesta

Povezavo s strežnikom smo vzpostavili s prostim programskim orodjem *Bitvise SSH Client*. Preko tega smo nalagali vse datoteke, ki smo jih potrebovali za postavitev portala *RIS*.



Slika 6: BitVise SSH Client

Najprej smo na strežnik naložili okolje *Wordpress* in datoteke, ki so vsebovale učna gradiva ter druge pripomočke za realizacijo opisane učne ure *Spoznavanje okolja Scratch*.



Slika 7: BitVise SFTP okno

S prvim dostopom na naslov <http://hrast.pef.uni-lj.si/ris/> smo po navodilih povezali Wordpress okolje s prej pripravljeno podatkovno bazo.

Spodaj morate vnesti podrobnosti o povezavi do podatkovne zbirke. Če o njih niste prepričani, kontaktirajte svojega spletnega gostitelja.

Ime podatkovne zbirke  Ime podatkovne zbirke, v kateri želite odpreti WP.

Ime uporabnika  Vaše uporabniško ime za dostop do MySQL.

Geslo  ...in geslo za dostop do MySQL.

Gostitelj podatkovne zbirke  Če lokalni gostitelj ne deluje, bi vam moral te podatke posredovati vaš spletni gostitelj.

Predpona tabele  Spremenite to polje, če želite v posamezni podatkovni zbirki zagnati več namestitev WordPressa.

Slika 8: Konfiguracija okolja Wordpress

Nato smo se v brskalniku prijavili v okolje Wordpress ter pričeli z urejanjem portala RIS.

## **7. NADALJNJE DELO**

V diplomskem delu smo predstavili realizacijo učne teme *Spoznavanje okolja Scratch*. V nadaljevanju imamo namen predstaviti tudi ostale tri teme, ki smo jih tu le opisali. To so: *Risanje po navodilih*, *Zaznavanje v Scratchu na primeru labirinta* in *Učenje programiranja preko zgodb in ugank*.

Za pridobitev povratne informacije o učinkovitosti spletnega portala in o uporabnosti že objavljenega učnega gradiva bomo ponovno povprašali učitelje, s katerimi bomo opravili intervjuje. Glede na njihove odgovore bomo temu primerno priredili oziroma popravili portal *RIS* in upoštevali želje učiteljev pri nadaljnjem razvijanju učnih gradiv. Na portalu bomo dodali funkcijo posodobitve že obstoječih gradiv s strani učiteljev.

## 8. ZAKLJUČEK

V diplomskem delu smo predstavili učne teorije, ki opisujejo konstrukcijo znanja učenca. To konstrukcijo pa je možno najbolje doseči z lastno aktivnostjo in vpletenostjo v okolje, ki je oblikovano tako, da nas uči novih znanj.

Da bi lahko sledili skoraj vsakodnevnim nadgradnjam programske opreme in hitremu razvoju tehnologije, moramo poznati in razumeti osnovne računalniške koncepte. Učenje programiranja, ki ga pedagoško uvrščamo med strategije reševanja problemov, pripomore k razumevanju teh konceptov.

Sprememba paradigme od zgolj uporabe tehnologije k razumevanju in kreiranju novih vsebin se odraža tudi v spremembah učnih načrtov računalniških predmetov v osnovnih šolah. Cilj je, da se učence opremi s programerskimi znanji prav tako pa z veščinami, ki jih bodo lahko uporabljali tudi na drugih področjih v življenju. Pri kurikularnih spremembah učitelji potrebujejo dodatno podporo, kot je pokazala empirična raziskava.

Iz intervjujev, ki smo jih izvedli med osnovnošolskimi učitelji, lahko zaključimo, da res potrebujejo dodatno izobraževanje in učna gradiva, kot smo na začetku diplomskega dela predvidevali. Prav tako so zainteresirani za izmenjavanje informacij in učnih gradiv preko spleta. V ta namen bomo nadaljevali z razvojem spletnega portala *RIS*, Računalniško izobraževanje s Scratchem.

## 9. LITERATURA

- About. ScratchEd.* (2014). Pridobljeno iz ScratchEd: <http://scratched.gse.harvard.edu/>
- Alice.org.* (2014). Pridobljeno iz Alice.:  
[http://www.alice.org/index.php?page=what\\_is\\_alice/what\\_is\\_alice](http://www.alice.org/index.php?page=what_is_alice/what_is_alice)
- Kodu - Microsoft Research.* (2014). Pridobljeno iz Kodu: <http://research.microsoft.com/en-us/projects/kodu/>
- Scratch 2.0 - Scratch Wiki.* (2014). Pridobljeno iz Scratch Wiki:  
[http://wiki.scratch.mit.edu/wiki/Scratch\\_2.0](http://wiki.scratch.mit.edu/wiki/Scratch_2.0)
- Ackermann, E. (2001). *Piaget's Constructivism, Papert's Constructionism: What's the difference?* Prevezeto 14. 8 2014 iz CONSTRUCTIVISM: USES AND PERSPECTIVES IN EDUCATION. CONFERENCE PROCEEDINGS. 1 & 2, str. 85-94. Geneva: RESEARCH CENTER IN EDUCATION.
- Blikstein, P. (brez datuma). *Seymour Papert's Legacy: Thinking About Learning, and Learning About Thinking | Transformative Learning Technologies Lab.* Prevezeto 14. 8 2014 iz <https://tltl.stanford.edu/content/seymour-papert-s-legacy-thinking-about-learning-and-learning-about-thinking>
- Constructing Kids, M. E. (2013). *Visual programming language - infograph and introduction.* Pridobljeno iz Constructing kids. Learningm playing and constructing with children.:  
<http://constructingkids.com/2013/05/15/vpl/>
- Gelman, R., & Baillargeon, R. (1983). *A review of some Piagetian concepts.* New York: Wiley.
- Glastovec, U. (2012). *pefprints.pef.uni-lj.si/1159/1/Ursa\_Glastovec\_Kodu\_FINAL.pdf.* Pridobljeno iz [http://pefprints.pef.uni-lj.si/1159/1/Ursa\\_Glastovec\\_Kodu\\_FINAL.pdf](http://pefprints.pef.uni-lj.si/1159/1/Ursa_Glastovec_Kodu_FINAL.pdf)
- Hunt, J. M. (1961). *Intelligence and experience.* New York: Ronald.
- Kozulin, A., & Presseisen, B. Z. (1995). Mediated learning experience and psychological tools: Vygotsky's and Feuerstein's perspectives in a study of student learning. *Educational Psychologist*, 67-75.
- Lajovic, S. (2011). *Scratch. Nauči se programirati in postani računalniški maček.* Ljubljana: Pasadena.



Ministrstvo za šolstvo, z. i. (2002).

[http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/os/devetletka/predmeti\\_izbirni/Racunalnistvo\\_izbirni.pdf](http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/os/devetletka/predmeti_izbirni/Racunalnistvo_izbirni.pdf). Pridobljeno iz

[http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/os/devetletka/predmeti\\_izbirni/Racunalnistvo\\_izbirni.pdf](http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/os/devetletka/predmeti_izbirni/Racunalnistvo_izbirni.pdf)

Ministrstvo za šolstvo, z. i. (2013). Pridobljeno iz

[http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/os/devetletka/program\\_razsirjeni/Racunalnistvo\\_izbirni\\_neobvezni.pdf](http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/os/devetletka/program_razsirjeni/Racunalnistvo_izbirni_neobvezni.pdf)

MIT. (2011). Pridobljeno iz

<http://scratched.media.mit.edu/sites/default/files/CurriculumGuide-v20110923.pdf>

MIT. (2014). *Scratch - Zamisli si, programiraj, deli*. Pridobljeno iz <http://scratch.mit.edu/>

MIT. (2014). *ScratchJr - About*. Pridobljeno iz ScratchJr: <http://www.scratchjr.org/about.html>

Prensky, M. (2001). *Digital Game-based Learning*. McGraw Hill.

Resnick, M. (2012). *Mother' Day, Warrior Cats, and Digital Fluency: Stories from the Scratch Community*. Pridobljeno iz <http://web.media.mit.edu/~mres/papers/mothers-day-warrior-cats.pdf>

Tharp, R. G., & Gallimore, R. (1988). *Rousing minds to life: Teaching, learning, and schooling in social context*. New York: Cambridge University Press.

Tomasello, M., Kriger, A. C., & Ratner, H. H. (1993). Cultural learning. *Behavioral and Brain Sciences*, 495-511.

Wikipedia. (2014). *Constructionism (learning theory)*. Pridobljeno iz Constructionism (learning theory):

[http://en.wikipedia.org/w/index.php?title=Constructionism\\_\(learning\\_theory\)&oldid=613246411](http://en.wikipedia.org/w/index.php?title=Constructionism_(learning_theory)&oldid=613246411)

Wikipedia. (2014). *Visual programming language*. *The Free Encyclopedia*. Pridobljeno iz Wikipedia. The Free Encyclopedia.:

[http://en.wikipedia.org/wiki/Visual\\_programming\\_language](http://en.wikipedia.org/wiki/Visual_programming_language)

Woolfolk, A. (2002). *Pedagoška psihologija*. Ljubljana: Educy.

Žist, D., & Oblak, I. (2004). Razvojna teorija Leva Vigotskega. *Socialna pedagogika*, 197-226.