Taksonomija alata namjenjenih učenju programiranja

Sinožić, Sebastian

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

SEBASTIAN SINOŽIĆ

TAKSONOMIJA ALATA NAMIJENJENIH UČENJU PROGRAMIRANJA

Završni rad

Pula, kolovoz, 2017.

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

SEBASTIAN SINOŽIĆ

TAKSONOMIJA ALATA NAMIJENJENIH UČENJU PROGRAMIRANJA

Završni rad

JMBAG: 0303046213, redoviti student
Studijski smjer: Informatika
Predmet: Programiranje
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija
Mentor: doc. dr. sc. Tihomir Orehovački

Pula, kolovoz, 2017.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani SEBASTIAN SINOŽIĆ, kandidat za prvostupnika informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

IZJAVA

o korištenju autorskog djela

Ja, SEBASTIAN SINOŽIĆ, dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom **"Taksonomija alata za učenje u programiranju"** koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.
Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

SADRŽAJ

Uvod **5**

1. Alati za učenje programiranja u okruženju Web 2.0 ..... **7**
1.1 Web 2.0 ..... **7**
1.2 Najpogodniji alati i aplikacije za učenje programiranja u Web okruženju **9**
1.2.1 Clou9 IDE 10
1.2.2 Ideone 11
1.2.3 JDoodle 12
1.2.4 Repl.it 13
1.3 Blog i wiki kao kolaborativni alati za učenje programiranja **14**
1.3.1 Blog 14
1.3.2 Wiki 15

2. Alati za učenje programiranja putem video igara **17**
2.1 Igre za učenje programiranja namijenjene djeci **19**
2.1.1 Tynker 19
2.1.2 Waterbear 20
2.1.3 RoboMind 21
2.1.4 Kodable 22
2.2 Igre za učenje programiranja namijenjene studentima **23**
2.2.1 CodeCombat 23
2.2.2 Code Hero 24
2.2.3 Code Hunt 25

3. Vizualni alati za učenje programiranja **26**
3.1 Alati za dijagram toka **28**
3.1.1 RAPTOR 29
3.1.2 Flowgorithm 30
3.2 Mini - jezici **31**
3.2.1 Guido van Robot 32
3.2.2. MSWLogo 33
3.3 Algoritmi za vizualizaciju **34**
3.3.1 ViSa 34

4. Skupina viših jezika za programiranje zajedno s alatima za pomoć učenju programiranja **36**
4.1 C++ i Verifikator **37**
4.1.1 C++ 37
4.1.2 Verifikator 37
4.2 Java i QAPlug **39**
4.2.1 Java 39
4.2.2 QAPlug 39
4.3 Phyton i Prospector **40**
4.3.1 Python 40
4.3.2 Prospector 41

5. Usporedba alata namijenjenih učenju programiranja **42**

Zaključak **45**

Literatura **47**
Popis slika **51**
Popis tablica **51**
Sažetak **52**
Abstract **53**

Uvod

Kao što nam i sam naslov teme sugerira, u ovom završnom radu izvršiti će se taksonomija alata namijenjenih učenju programiranja. Taksonomija je znanstvena disciplina koja na temelju sličnosti i razlika, taksonomske jedinice kategorizira i razvrstava u skupine ili grupe. Tako i u ovom slučaju, kao što kaže naslov, u zadatku samog završnog rada, izvršiti će se kategorizacija i svrstavanje alata za učenje u programiranju u četiri osnovne skupine, odnosno područja primjene, a sve na temelju njihove namjene, korištenja i primjene u učenju programiranja. Skupine alata za učenje programiranja podijeljene prema njihovoj namjeni, koje se spominju u nastavku jesu slijedeće:

1. Alati za učenje programiranja u okruženju Web 2.0,
2. Alati za učenje programiranja putem video igara,
3. Vizualni alati za učenje programiranja i
4. Skupina viših jezika za programiranje zajedno s alatima za pomoć u učenju programiranja.

U prvoj skupini obraditi će se WEB servis kao mjesto na kojemu je moguće steći predznanja i vještine, a koje nam omogućuju shvaćanje i savladavanje osnova za početak učenja programiranja, zajedno s srodnim alatima koji nam omogućuju aktivnosti unutar samog servisa.

U skupini koja obrađuje učenje programiranja kroz igrice obraditi će se skupina alata koji se mogu koristiti za učenje u programiranju unutar samih igrica. Obzirom da su potencijalni programeri u najvećem postotku upoznati s igranjem igara, načinom igre i ciljevima igre kao dobri poznavatelji takve tematike, zasigurno je to jedan od važnih čimbenika koji će potaknuti potencijalnog programera da se dodatno zainteresira i angažira u stjecanju znanja i vještina na zabavan i kreativan način, upravo kroz alate koji su obrađeni u ovoj skupini.

Unutar skupine vizualnih alata za učenje u programiranju, obrađeni su alati koji se odnose na prelazno razdoblje, odnosno na period između faze učenja programiranja kroz igrice i faze viših programskih jezika. Obzirom da je izravan prijelaz gotovo nemoguć, a bio bi kontradiktoran ovom završnom radu, unutar ove

skupine obrađeni su vizualni alati koji će potencijalnog programera usmjeriti putem vizualnih alata u same početke "pravog" programiranja.

U zadnjoj skupini koja zaokružuje zadatak ovog završnog rada, obraditi će se alati za učenje u programiranju, koji se prvenstveno fokusiraju na studente kao buduće članove akademske zajednice, a koji stasaju kao programeri i zajednički im je cilj naučiti znanja i vještine programiranja kao svog životnog poziva i usmjerenja. U prikazu se obrađuje inicijacija programiranja, osobito važan prelazak s početničkog nivoa znanja na savladavanje i učenje na tzv. višim alatima za programiranje, koji nas nakon usvajanja vještina rada na istima, prate u daljnjem životu i radu kao najosnovniji resurs znanja i podloga su za daljnje životno usavršavanje.

Kroz naprijed navedene skupine alata za programiranje, potencijalnog korisnika, odnosno čitatelja, upućuje se u mogućnost izbora alata za programiranje bez da se nepotrebno luta i započinje s svladavanjem znanja i vještina za koje bi kasnije utvrdili da nisu u potpunosti ono što je tražio, kao što se zna dešavati u praksi. Na ovaj način grupiranja potencijalni korisnik, odnosno čitatelj, izravno može pristupiti učenju programiranja kroz skupine alata, sve ovisno od namjene i želje koja ga interesira.

1. Alati za učenje programiranja u okruženju Web 2.0

1.1 Web 2.0

Web 2.0 se javlja kao nova generacija značajnog internetskog servisa 21. stoljeća, World Wide Web-a. Iako sam naziv Web 2.0 govori da se radi o novom standardu, taj termin se ne odnosi na tehničko poboljšanje, već na promjene u načinu korištenja Weba, gdje se web koristi kao društvena platforma. Korisnici više nisu samo čitatelji sadržaja i primatelji informacija, već aktivno sudjeluju u njihovom stvaranju, izmjenjivanju i prenošenju istih.

Web 2.0 kao logičan slijed nastao je od Web-a 1.0 radi velikog pritiska tržišta i korisnika, a sve radi sve većeg korištenja Interneta.

Prema O'Reilly (2009.,str. 22) razmišljanju Web 2.0 možemo podijeliti u četiri osnovne kategorije:

- **Aplikacije III. razine** – aplikacije postoje isključivo na internetu te njihov porast i poboljšanje ovise o broju aktivnih korisnika. Primjeri su: eBay, Amazon, Skype, Discord
- **Aplikacije II. razine** – imaju minimalne funkcionalnosti u offline radu, ali punu funkcionalnost postižu online. Primjeru su: Facebook, Flicker, Twitter
- **Aplikacije I. razine** – imaju veliki sklop funkcionalnosti u offine načinu rada, ali bitne značajke postižu u online načinu rada. Primjeri: Office 365, iTunes
- **Aplikacije nulte razine** – aplikacije koje imaju jednake funkcionalnosti u mrežnom i izvan mrežnom načinu rada. Primjeri: Google Maps, ViaMichelin, Sygic

Ideja Web 2.0 bila je izbaciti nedostatke Weba 1.0 koji je predisponirao isključivo komunikaciju u jednom smjeru, gdje je korisnik bio samo čitatelj sadržaja na web stranicama. U središtu pažnje su korisnici kojima je omogućeno pojednostavljeno korištenje, ali i participacija u stvaranju, ažuriranju, brisanju i prijenosu informacija. Dakle, u Webu 2.0 omogućena je dvosmjerna komunikacija između korisnika i poslužitelja, iz čega proizlazi da korisnik više nije samo u svojstvu čitatelja, već postaje aktivni sudionik u stvaranju i distribuciji informacija.

Sudjelovanjem većeg broja aktivnih korisnika u stvaranju sadržaja, podaci se konstanto skupljaju, čime se baza podataka povećava, obogaćuje te se posljedično javljaju novi načini korištenja istih.

Trenutno se najučestalije koriste tehnologije u sklopu Web 2.0: blogovi, wikiji, podcasti, društveno umrežavanje, spajanjem informacijskih sadržaja iz brojnih izvora, direktna razmjena informacijskih sadržaja (engl. Peer-to-Peer, P2P) i označavanje informacija (engl. Tagging).

Tip usporedbe	Web 1.0	Web 2.0
Preferirani način uporabe	Čitanje	Pisanje
Predvodnici	Tvrtke ili korporacije	Sami korisnici
Arhitektura	Klijent – server	Peer – to – Peer
Najznačajniji standard	HTML	XHTML, XML, CSS
Stranice pojedinaca	Osobne stranice	Blogovi
Način informiranja	Portali	RSS (više izvora)
Organizacija podataka	Taxsonomy (grupiranje u kategorije)	Tagovi (korisnici sami biraju ključnu riječ za grupaciju)
Tehnologija prijenosa podataka	Žičana	Bežićna
Vlasništvo	U vlasništvu pojedinaca	Dijeljeno (sharing)
Način kapitalizacije	Prva javna ponuda dionica (IPO)	Prodaja prava (trade sales)
Tvrtka – tehnološki predvodnik	Netscape	Google
Način interakcije	Web forme	Web aplikacije
Preuzimanje sadržaja sa drugih stranica	Screen scraping (sa stranice)	API, RSS, XML
Veza na Internet i način razmišljanja	Spora putem analognog modema i dial-up	Širokopojasna, stalno online
Troškovi o kojima se razmišlja	Cijena hardware-a	Cijena propusnosti podataka

Slika 1. Razlike između Web 1.0 i Web 2.0 (Codeacademy, 2013)

1.2 Najpogodniji alati i aplikacije za učenje programiranja u Web okruženju

U odnosu na nekadašnje web mogućnosti, dolaskom Web 2.0 kao smjera u World Wide Web tehnologiji, omogućena je svojevrsna tehnološka socijalizacija koja potencijalnim korisnicima omogućuje sudjelovanje u stvaranju sadržaja weba, ali i podrazumijeva interaktivnu dvosmjernu komunikaciju, kako između korisnika i računala, tako i između jednog ili više korisnika međusobno, čime korisnik postaje aktivni sudionik tog procesa socijalizacije.

Najvažnija razlika između WEB-a 2.0 i svog prethodnika, očituje se u tome što se nekadašnja radna površina računala „preselila“ na Web 2.0 okruženje. Svi alati koje smo nekada morali sadržavati na vlastitom računalu, sada se nalaze na nekoj od platformi weba, što je uvelike omogućilo lakše korištenje, nadogradnju i ažuriranje postavki, smanjilo potrebu za prostorom za pohranu svih tih alata na računalu te niz drugih pogodnosti. Sve podatke koje koristimo, stvaramo ili tražimo lako je dijeliti, oblikovati, ali i učestvovati u izradi istih.

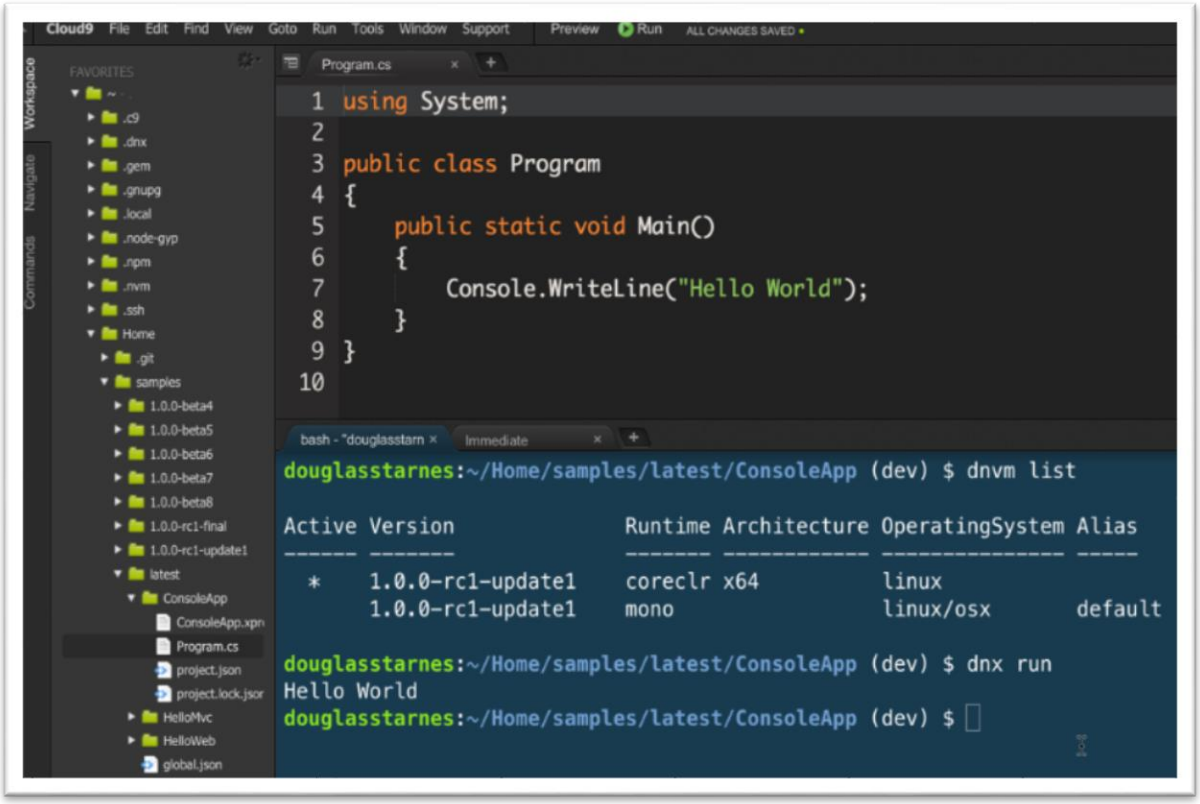
Upravo ta mogućnost interakcije u Web 2.0 okruženju, vrlo je pogodna za učenje programiranja. Možda ne toliko izravno s naglaskom na učenje, ali navedena socijalna komponenta omogućuje nam kolaboraciju i interakciju s drugim korisnicima, a posljedično i mogućnost za razmjenu informacija, savjeta, pitanja, znanja ili drugih korisnih oblika u učenju.

Osnovni pojam koje se koristi u razvoju softvera je IDE (Integrated Development Environment), koji omogućavanje pisanje, prevođenje i testiranje programa.

Sukladno kriterijima koje su Škorić, Pein i Orehovački (2016) naveli u svojem radu, a kriteriji su: da alat podržava C++ programski jezik, da alat bude besplatan, da je alat otvorenog izvora – koda (open source) te da alat dopušta dijeljenje koda,u nastavku je prikazano nekoliko primjera, kojima je prikazano koji su web IDE alati i aplikacije najpogodniji za učenje programiranja u web okruženju.

1.2.1 Clou9 IDE

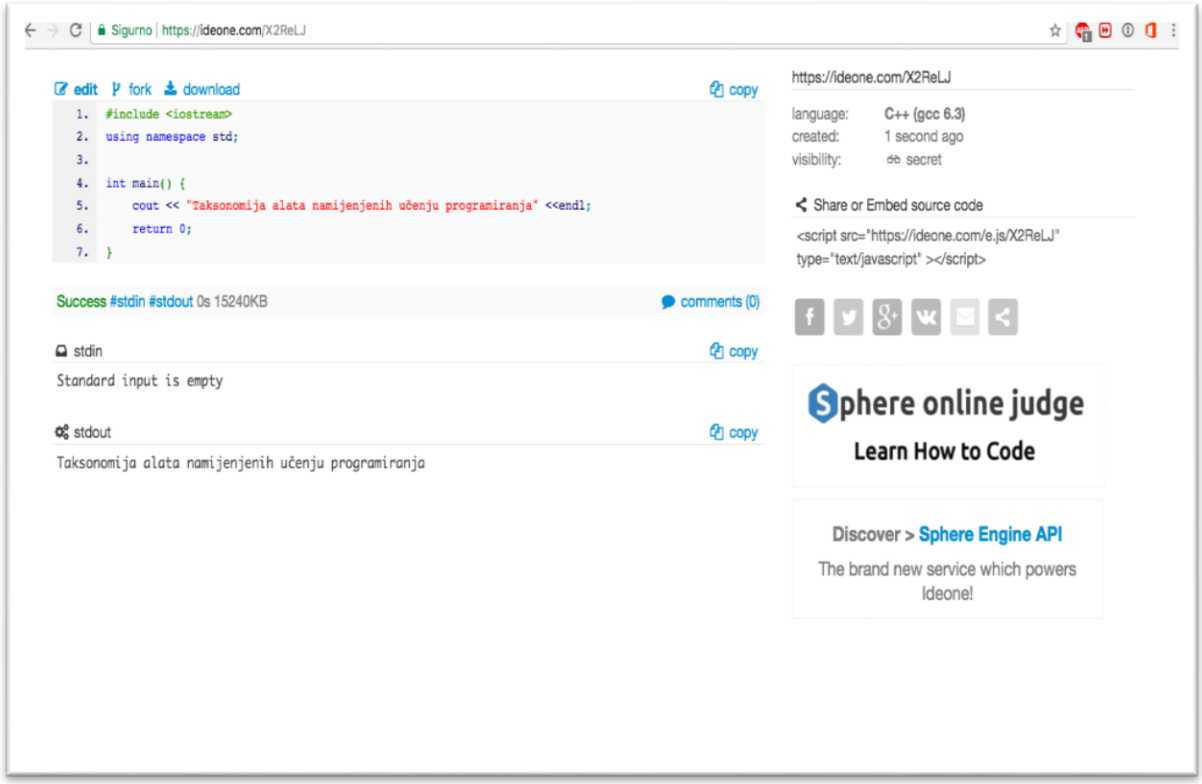
Clou9 IDE je online integrirano programsko okruženje koje podržava razne programske jezike, uključujući: C, C++, Ruby, PHP, Javascript i druge. Korisnici mogu odmah krenuti s radom na konfiguriranom radnom prostoru, komunicirati i surađivati s ostalim korisnicima te pokrenuti i koristiti razne web značajke kao što su pregled stvorenog sadržaja u realnom vremenu i testiranje kompatibilnosti preglednika.



Slika 2. Clou9 IDE okruženje

1.2.2 Ideone

Ideone je online alat za kompiliranje koda koji omogućuje sastavljanje izvornog koda i njegovo izvršavanje online na više od 60 programskih jezika. Nakon izvršenja koda korisnik može isti dijeliti na niz društvenih mreža kako bi dobio povratnu informaciju od ostalih korisnika, s naglaskom na savjetodavnu komponentu; što može poboljšati ili pak ispraviti. Ideone sadrži velik broj popularnih jezika kao što su: C++, Ruby, Javascript, SQL, PHP, Python i drugi.



Slika 3. Ideone online kompilator

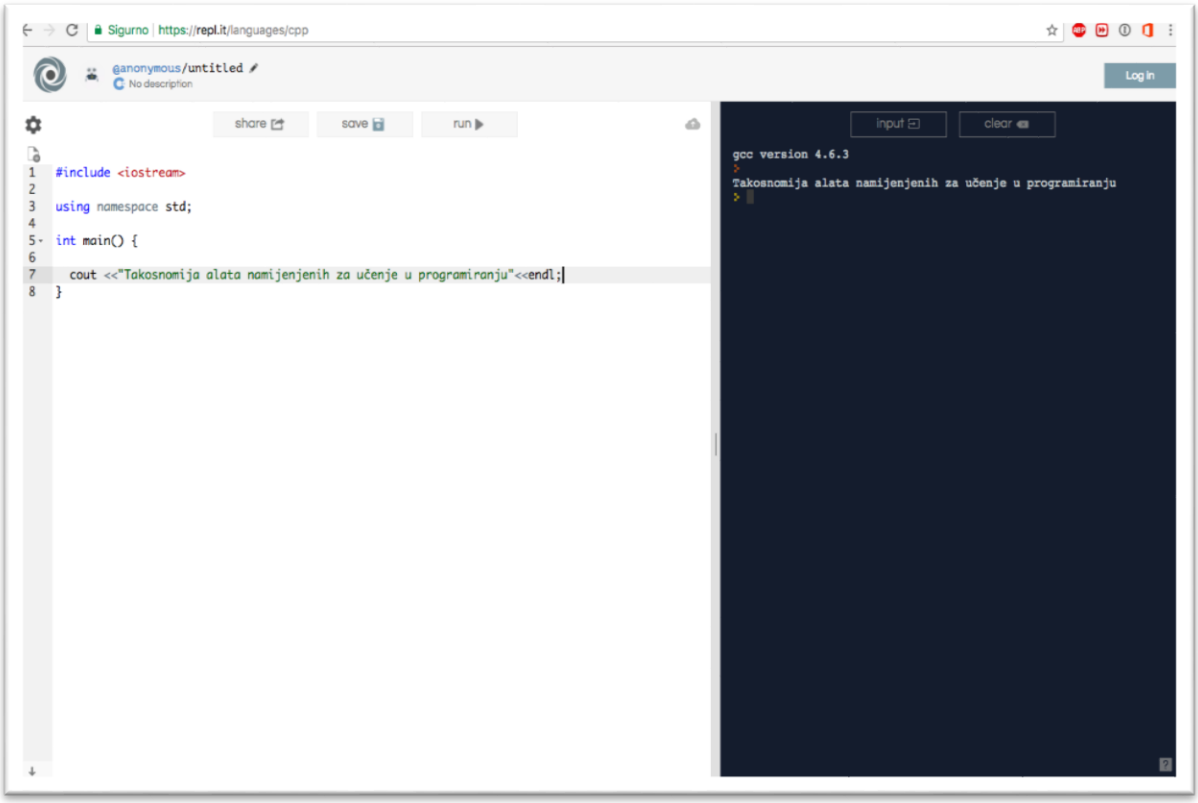
1.2.3 JDoodle

JDoodle je alat na mreži koji služi za sastavljanje i izvršavanje programa. Sadržava širok broj jezika kao što su: C, C++, PHP, Python i drugi. Glavni odlika, a posljedično i cilj ovog alata je; na brz i jednostavan način stvoriti, izvršiti, spremiti ili podijeliti kod, što će eliminirati potrebu otvaranja novog projekta na izvan mrežnom okruženju. JDoodle ima mogućnost implementacije na blog / web stranice, kao i direktnog izvršavanja koda sa istih.

Slika 4. JDoodle online kompilator

1.2.4 Repl.it

Repl.it je jednostavan, ali vrlo moćan online IDE kompilator. Alat pruža mogućnost spremanja sesije, na način da generira vezu s kojom se kasnije može pristupiti na točnu poziciju gdje smo prethodno stali. Prednost ovog alata je mogućnost korištenja na tablet i mobilnom uređaju uz širok pojas popularnih programskih jezika kao što su: C++, C, PHP, Java, Python i drugi. Najvažnija značajka ovog alata jest u tome što postoji inačica za profesore putem koje mogu kreirati virtualne učionice te učenicima/studentima zadavati zadatke.



Slika 5. Repl.it online programsko okruženje

1.3 Blog i wiki kao kolaborativni alati za učenje programiranja

1.3.1 Blog

Blog (eng. Weblog), kao alat, je združeni naziv za online objave na Webu u obliku dnevnika.

Na blogu se objavljuju informacije u tekstualnom obliku, niz slika, zvukovni isječci, razni video zapisi, a sve navedeno najčešće je taksativno objavljeno po kronološkom redu.

Stvaranje i objavljivanje sadržaja vrši se kroz standardni web preglednik i ne zahtijeva znanja za programiranje, što omogućuje svakoj osobi da podijeli svoja razmišljanja, znanja i iskustva.

Blogovi su zamišljeni kako bi blogeri svojim pratiteljima iznijeli vlastita razmišljanja, stavove ili uvjerenja, a većinom su pisani konverzacijskim ili slobodnim stilom. Svaki blog omogućava komentiranje kojom prilikom pratitelji mogu iznijeti svoje replike, komentare i raspravljati s drugim pratiteljima.

Uobičajeno je na blogovima ostaviti mogućnost pridruživanja oznake za kategoriziranje i lakše snalaženje pratitelja, tako da se pratitelji mogu pretplatiti na blog putem RSS tehnologije, a sve radi blagovremenog praćenja novih objava.

Postoje individualni i kolaborativni blogovi. Mogu biti povezani u grupe, tematski ili nastavno na domenu u kojoj su smješteni – tzv. blogosfera.

Blogovi su danas široko rasprostranjeni pa se tako mogu koristiti kao online dnevnici, stranice za osobne potrebe ili informiranje, kao web stranica za pružanje informacija, a zajednička im je osobina što se mogu vrlo lako ažurirati. Također služe kao odlična sredstva za informiranje u poslovnim krugovima.

Ono što je bitno za napomenuti u ovom završnom radu, nastavno na temu rada, uvelike je sadržano u istraživačkom radu grupe autora Bubaš, Ćorić i Orehovački (2012), u kojem grupa autora stavlja drugačiji naglasak na ovaj alat. U navedenom istraživačkom radu, u suradnji sa studentskom zajednicom, provedeno je istraživanje u obliku korištenja Wordpress blog sustava kao referentnog alata za vođenje bilješki prilikom predavanja. Studenti su zamoljeni da tijekom predavanja bilježe bitne anotacije s nastave u obliku zapisa na blogu. Uz primarni zadatak vođenja bilješki, studenti su imali zadatak implementirati sadržaje na stranicu bloga, koristeći i druge razne alate iz okruženja Web 2.0 tehnologije (fotografije,

videozapise i sl.). Na kraju nastavne cjeline završno su morali izraditi esej kojim su prikazali znanja, vještine, koje su koristili i stekli tijekom predmetne nastavne cjeline. Nastavno na temu ovog rada, može se zaključiti da Worpress blog tehnologija može studentima pomoći pri učenju, jednim dijelom kao samostalna cjelina, a drugim dijelom kao sadržaj na webu kojeg mogu koristiti i ostali studenti odnosno korisnici.

1.3.2 Wiki

Wiki je alat koji je predisponiran za suradnju i zajedničko pisanje te je izvrstan primjer iskonskih načela na kojima se i zasniva WEB 2.0, a to su prvenstveno otvorenost i sloboda načela te sveprisutna kolektivna inteligencija.
Najveći naglasak sadržan je u ideji da se što većem broju korisnika omogući sudjelovanje u sukreiranju zajedničkog sadržaja, a nastavno na uvod, također i u uređivanju, korekciji, nadogradnji ili pak brisanju već postavljenih sadržaja. Kreiranje i uređivanje takvih sadržaja je izuzetno jednostavno, a vrši se putem standardnog web preglednika.

Wikiji su na osnovu svojih osobina vrlo pogodni te se mogu koristiti za razne kolaboracijske projekte, izmjenu razmišljanja i razvoj ideja, a sveobuhvatnu primjenu pronašli su u poslovnom svijetu kao rješenje za upravljanje znanjem te se iz tog razloga koriste i u obrazovanju. Jednostavnošću i niskim troškovima održavanja izuzetno su pogodni za obrazovno okruženje s malim proračunom. Wikipedija se oslanja na postulate otvorenosti, zajedničke suradnje, kolektivnog znanja i jednostavnosti, odnosno na same temelje wikija.

Međutim, iako osnovni temelji wikija, kao što su otvorenost, mogućnost rasprave i sudjelovanja velikog broja sukreatora i korisnika omogućili da nastane jedan ovakav resurs izuzetne važnosti, ipak ponekad dođe do objave netočnih podataka, što je jedan od većih nedostataka ovakvog alata. Pozitivno je što u sukreiranju učestvuje veći broj korisnika i kreatora pa se takvi nedostaci brzo otklone.

Kao primjer koji ćemo u nastavku prikazati, također jer obrađen u znanstvenom radu od strane grupe autora Bubaš, Ćorić i Orehovački (2012), u kojem su autori također koristili studentsku zajednicu kao suradnike za navedeno istraživanje. Studentima je zadan višestupanjski zadatak kroz cijelu nastavnu cjelinu, u kojoj je bilo neophodno koristiti veći broj alata iz Web 2.0 tehnologije (npr. Google Dokumenti – kolaborativno pisanje, Delicious – stvaranje on-line oznaka, Mindomo, -

stvaranje umnih mapa, Gliffy – izrada dijagrama toka, Masher – video podcasting te Snipt – za kolaborativno programiranje). Rezultat korištenja svih navedenih alata je sadržaj koji je u zadnjoj fazi zadatka potrebno implementirati na wiki stranicu koju su prethodno morali izraditi. Wiki stranica je u konačnici sadržavala sve faze izvršavanja zadatka, zajedno s poveznicama prema rezultatima i svim korištenim alatima unutar zadatka. Putem web preglednika ostali studenti mogli su pristupiti i pregledavati sadržaj i kompletnu kronologiju izvršavanja zadatka.

Upravo na taj način, a sukladno zaključcima iz gore navedenog znanstvenog rada, možemo ustvrditi da je više od 2/3 studenata ustvrdilo da je ovakav način prezentacije materijala vrlo efektivan, puno prihvatljiviji u odnosu na konvencionalne materijale te da će im rezultati sa wiki stranica pomoći u daljnjem učenju i radu, dok je ukupna ocjena iznosila 3.63, što je vrlo visoka ocjena.

2. Alati za učenje programiranja putem video igara

U ovom ćemo dijelu završnog rada prikazati koji alati pomažu učenju kroz igru i na koji način, a kriteriji koji su primijenjeni jesu: podjela igara prema uzrastu te ekonomski kriterij (da su igre besplatne).

Općenito je uvriježeno mišljenje da je programiranje kao postupak, nešto zamršeno, komplicirano pa samim time i dosadno, nepristupačno i sl. Takva percepcija učenika i studenata, jedan je od većih razloga straha od programiranja, zaziranja od samog početka programiranja pa samim time i potiskivanje ikakvih pomisli da se krene programirati.

Kako bi djeci i studentima približili, demistificirali i na kvalitetan način prezentirali početke učenja u programiranju, krenuti ćemo od osnovnih dječjih interesnih sfera, a to su naravno igrice, jer su upravo igrice ono što nas prati cijelog života, kao uvijek lijep i nostalgičan podsjetnik na djetinjstvo.

U ovom radu prikazati ćemo igrice podijeljene prvenstveno prema dobnoj raznolikosti, odnosno igrice namijenjene djeci u osnovnom školstvu te djeci u srednjoškolskom obrazovanju zajedno s studentima koji započinju studije.

Cilj alata koji kroz igrice predočava i potiče djecu na početak igranja – programiranja jest demistifikacija programiranja kao nečeg teškog i kompliciranog i približavanje programiranja kroz dobro smišljene igre koje potiču znatiželju i potiču nastavak igranja, odnosno programiranja. Na taj način, kada se pravovremeno uputi djecu kroz igrice u mogućnost programiranja (iako kroz zabavu), u konačnici se stječe predispozicija i velika vjerojatnost da će i u nastavku školovanja nastaviti programirati bez straha i zaziranja te da će takvu problematiku percipirati kao nešto zabavno i korisno.

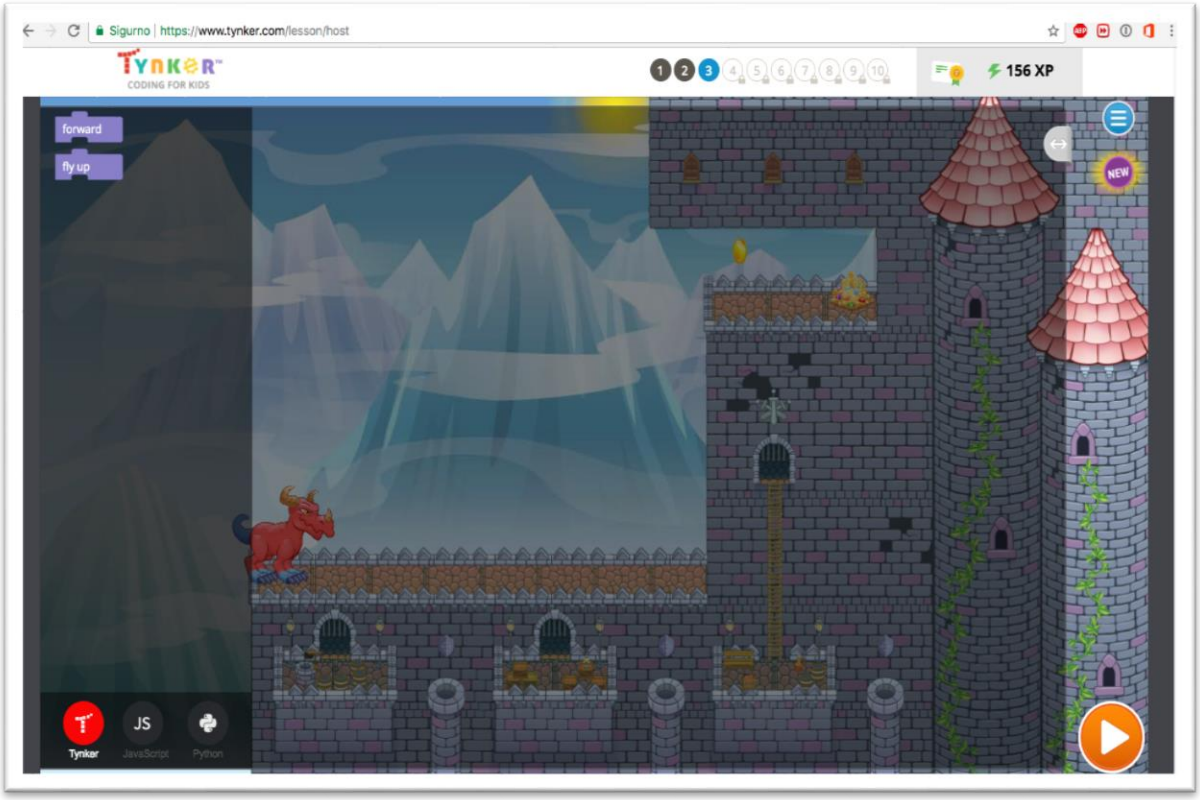
Dakle, upravo su igrice, počevši od dječje dobi od kada su one relativno jednostavne, zamašnjaci koji potiču maštu ali i znatiželju djeteta za drugim kvalitetnijim i složenijim igricama. Ukoliko je nastavni program kvalitetno složen da na isti način potiče znatiželju i maštu djece, tada će djeca u vrlo kratkom razdoblju savladati predviđena znanja i vještine, jer ih to asocira na igranje igrica. Iste karakteristike se mogu primijeniti na sve razine školovanja pa tako i na studente, što su na sličan način Orehovački, Babić (2015) primijenili u svojem istraživanju grupe studenata.

Što se događa s istim pristupom kod studenata? Poznato nam je da je svrha igranja pobjeda, ili bilo koji drugi završetak igre koji donosi zadovoljstvo, sreću, ushićenje ili neki drugi sličan osjećaj, a upravo je baš to i osnovni cilj igranja. Kod studenata, odrastanjem i sazrijevanjem, taj se cilj pretvara u zadovoljstvo postignutim rezultatima, a benefit je isti kao i koda mlađih uzrasta; na zabavan i pristupačan način, ali s bitno ozbiljnijom tematikom igre, postigli smo zadani cilj da se studenti kroz takvu igru oslobode straha od programiranja, da shvate da je programiranje i zabavno i jednostavno te da prihvaćajući takve igre steknu samopouzdanje za pristup još većim izazovima bez predrasuda i sumnji u vlastito samopouzdanje.

2.1 Igre za učenje programiranja namijenjene djeci

2.1.1 Tynker

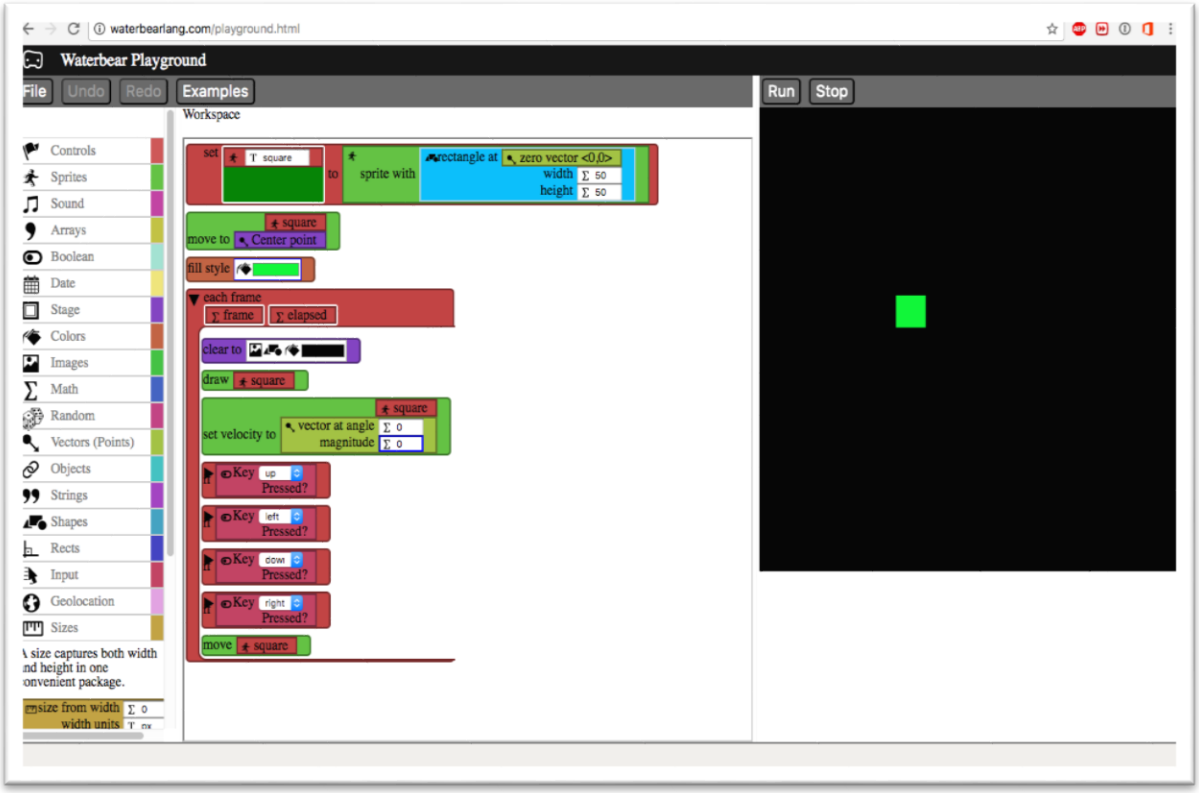
Tynker je sustav učenja programiranja namijenjen djeci kako bi kroz motivaciju prevodili svoje ideje u igre i projekte te sve to prenosili na aplikacije koje se nalaze na mreži. Tynker koristi jednostavan vizualni programski jezik u kojem nije potrebna programska sintaksa već samo spajanje kombinacija blokova kodova kako bi se određene radnje izvršile. Igra je besplatna, uz uvjet da se korisnik besplatno registrira.



Slika 6. Tynker - igra za najmlađe

2.1.2 Waterbear

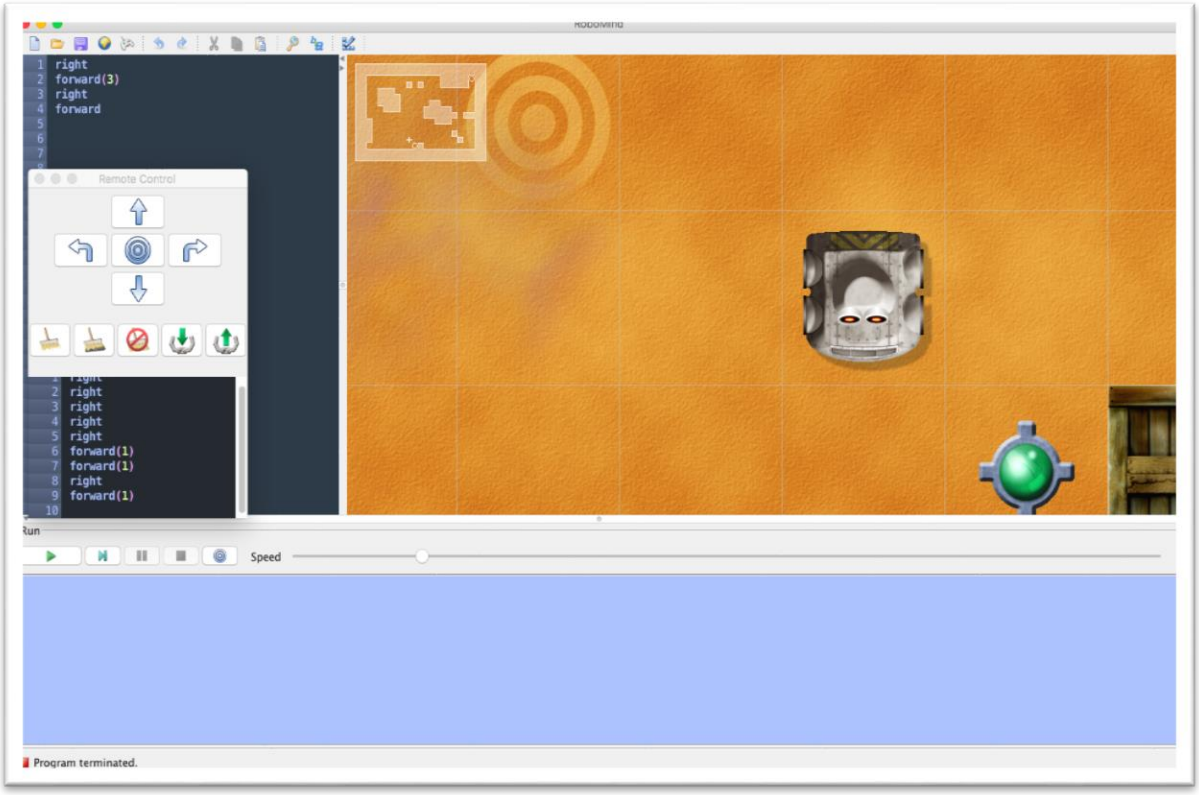
Waterbear je praktičan i besplatan alat za programiranje namijenjen djeci koji koristi metodu povlačenja i spuštanja (eng. drag and drop). Waterbear koristi vizualni programski jezik, što znači da nije potrebno učenje programske sintakse. Djeca mogu stvoriti novu datoteku ili pak pogledati primjere drugih korisnika te izmjenjivati njihove projekte s ostalima, kako bi bolje shvatili funkcionalnosti. Svaki element ima opis kako bi djeca lakše shvatila funkcionalnosti istih.



Slika 7. Waterbear igra spajanja blokova

2.1.3 RoboMind

RoboMind je alat za djecu koji koristi vlastiti jezik ROBO. Vrlo je jednostavan i ne zahtjeva predznanje o programiranju. Glavni cilj je pokretati i voditi robot kroz dvodimenzionalni svijet te odraditi zadane zadatke. Zadaci mogu biti obavljeni putem jednostavnih tipki koje ukazuju na smjer kretanja ili putem jednostavnog jezika ROBO. Alat je besplatan 365 dana ako se korisnik registrira te podržava Windows i Mac platformu.

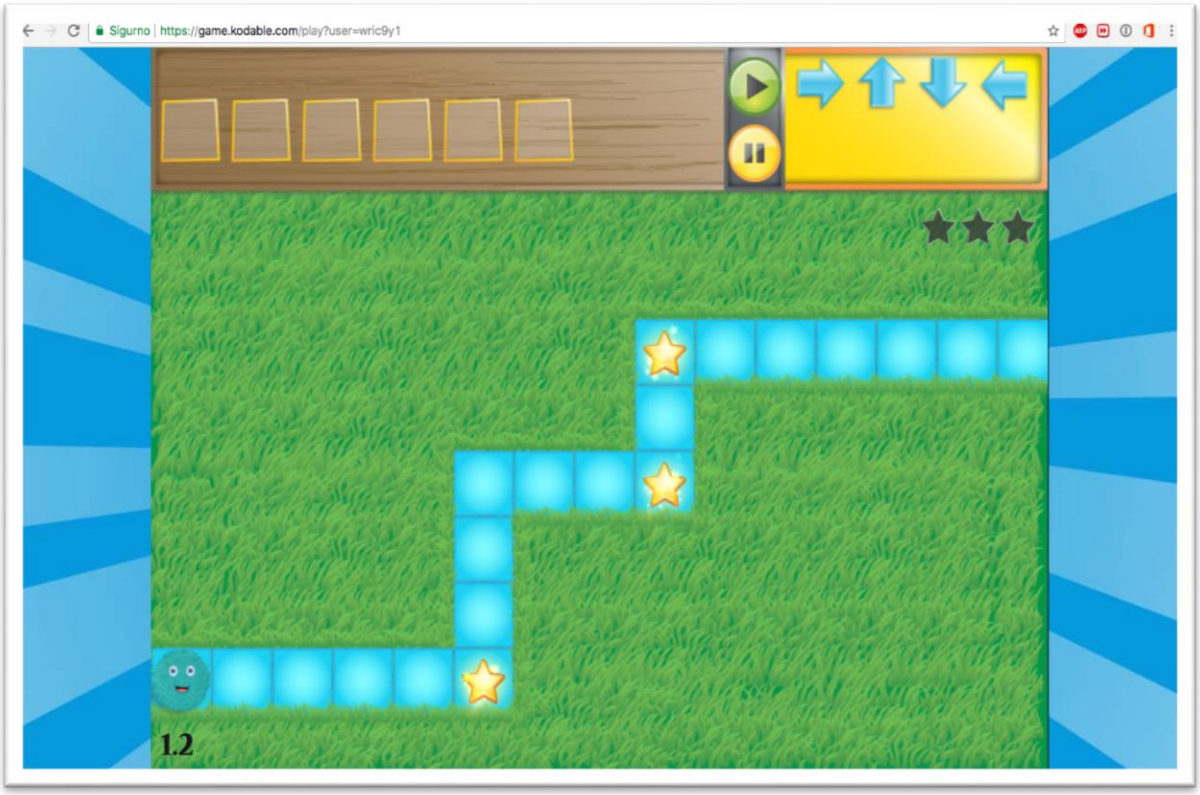


Slika 8. RoboMind igra pomicanja tenka

2.1.4 Kodable

Kodable je besplatna online aplikacija koji se može koristiti putem Web preglednika. Dizajniran je posebno za djecu u dobi od pet ili više godina, tako da mogu naučiti osnove programiranja igranjem uz pomoć malih instrukcija. U nastavku ćemo navesti najvažnije značajke ove aplikacije koje djeca mogu razviti igrajući:

- Logika i vještine rješavanja problema,
- Slijed dešavanja događaja,
- Uvjetne izjave – "Ako je to točno, to će se dogoditi",
- Petlje – ponavljanje skupa naredbi.



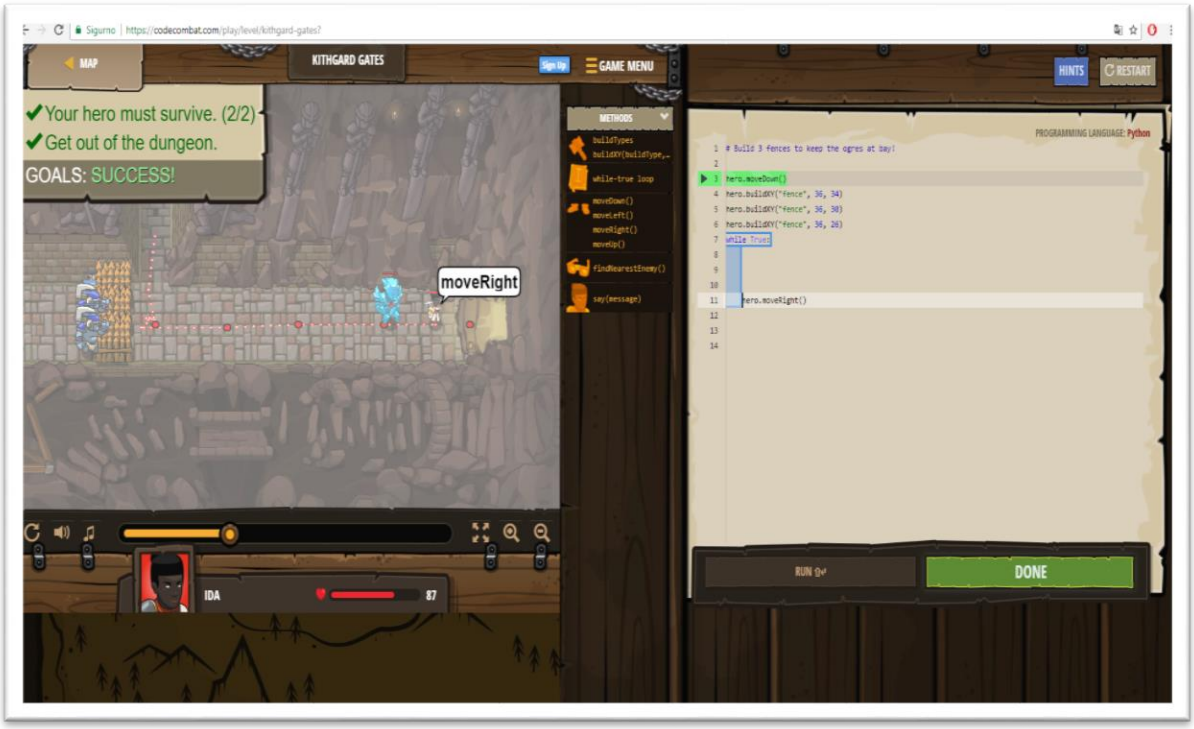
Slika 9. Kodable igra za djecu

2.2 Igre za učenje programiranja namijenjene studentima

2.2.1 CodeCombat

CodeCombat je video igra namijenjena učenju programiranju kroz web preglednik u jeziku Python. Stvorena je 2013. godine u San Francisku. Igra se sastoji u tome da korisnik kroz pisanje Python koda mora prolaziti različite nivoe igra. Nivoi se sastoje u tome da se korisnik mora kretati, izbjegavati ili pak napadati protivnike, kako bi riješio problematiku samog nivoa. Na početku nivoi su prilično lagani, te samim napretkom igre postaju sve teži pa tako i samo korištenje koda (korištenje petlji) postaje kompleksnije. U nastavku, navesti ćemo nekoliko značajki ove igre:

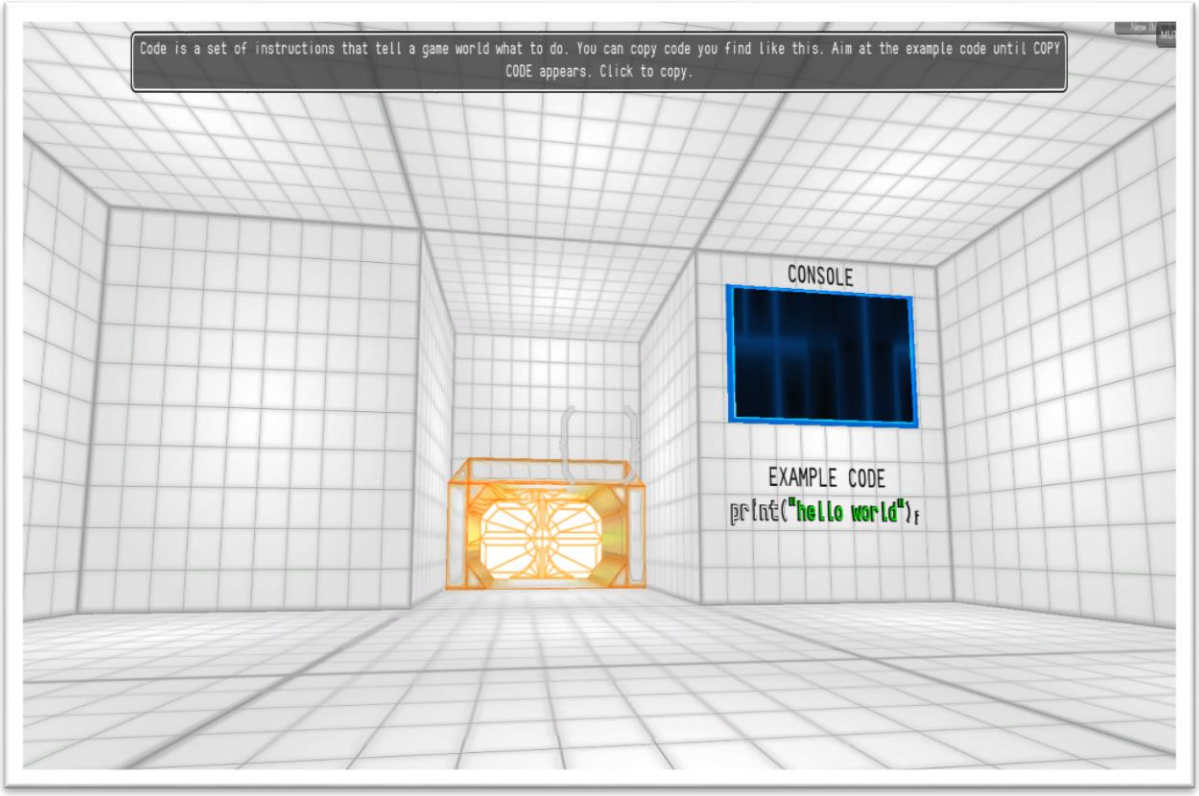
- Širenje osnova koda kao što su varijable, petlje, naredbe grananja (eng. if-else) i drugih metoda,
- Pružanje povratnih informacija kako bi korisnik bolje razumio logiku koda,
- Upoznavanje korisnika s pravilnim pisanjem sintakse i kodiranjem u obliku odabranog jezika,
- Priprema korisnika za kasnije implementiranje koda kroz razne aplikacije.



Slika 10. CodeCombat igra pomoću Python sintakse

2.2.2 Code Hero

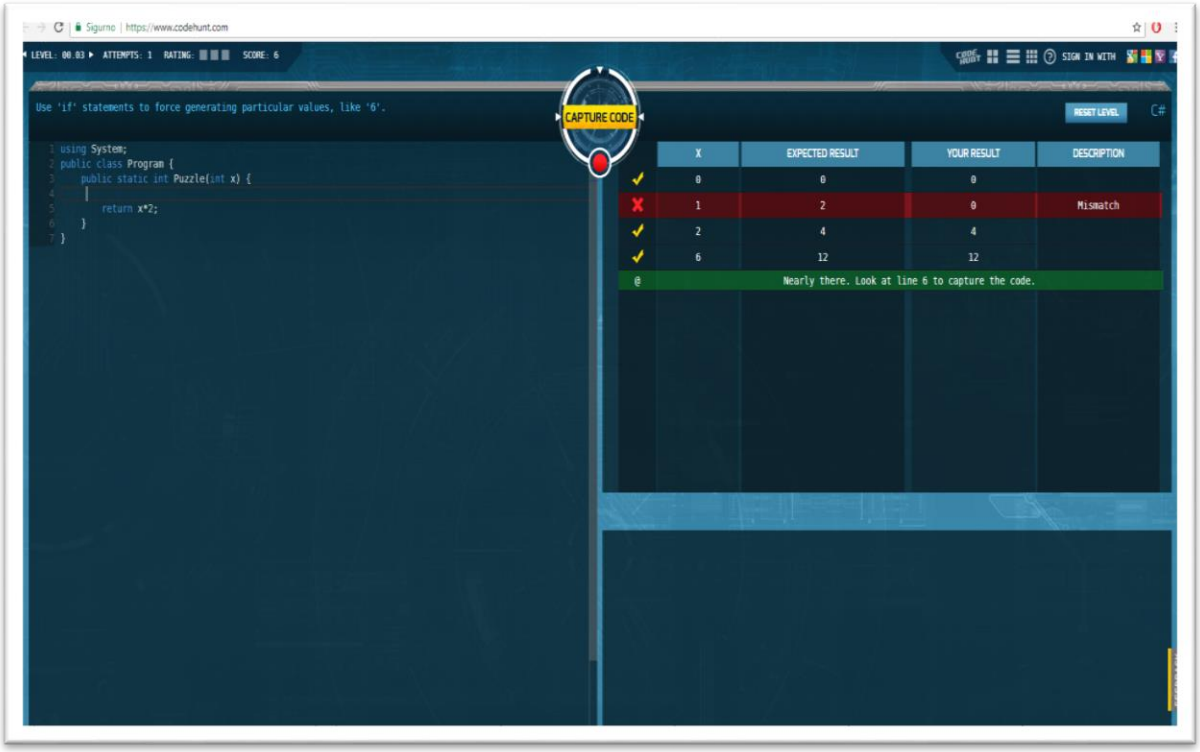
Code Hero je edukacijska video igra u kojoj sami korisnici igrajući izmijenjaju postojeću igru. Korisnici mogu smanjivati, rotirati te micati postojeće objekte kako bi prešli određeni nivo. Na težim razinama korisnici moraju u ugrađenom Unity editoru izmjenjivati vrijednosti objekata kako bi mogli otići na više razine. Igra je podržana za Mac i Windows platformu.



Slika 11. Code Hero igra unutar igre

2.2.3 Code Hunt

Code Hunt je igračka platforma za kodiranje u svrhu natjecanja i vježbanja programskih vještina. Igrica se izvršava u bilo kojem web pregledniku, a bazirana je na jezicima Java i C#. Kroz napredak razina korisnici se upoznaju s aritmetičkim operatorima, petljama, uvjetnim izjavama, nizovima, algoritmima za pretraživanje i drugih.



Slika 12. CodeHunt igra programske logike

3. Vizualni alati za učenje programiranja

U galopirajućem razvitku sveopće tehnologije, svjedoci smo izuzetno brzog razvitka uređaja i strojeva, kako za osobnu uporabu, tako i u svim granama industrije i djelatnosti. Kao posljedica takovog uznapredovanog razvitka, javlja se potreba za aktivnim učenjem i praćenjem napretka tehnologije, ukoliko želimo suvereno i kvalitetno koristiti ponuđenu tehnologiju na tržištu. Napredak je vidljiv na svakom koraku, a potreba za edukacijom potrebna nam je za; korištenje pametnih telefona, korištenje uređaja u javnom sektoru (porezna uprava, banke i bankomati, javne ustanove, javni prijevoz i sl.), korištenje kućanskih aparata i uređaja, korištenje pametnih funkcija u automobilima, korištenje prednosti koje nam omogućuje implementacija sustava pametnih kuća, korištenje dlanovnika te jednostavnijih računala pa sve do profesionalnog korištenja računala u profiliranim gospodarskim subjektima kao finalni proizvod istih.

Za sve navedeno, javlja se potreba programiranja svih tih uređaja i struktura; od osnovnog daljinskog uređaja za neki od navedenih medija pa sve do izrade programa za velike industrijske pogone.

Uvođenjem raznih aplikacija u javni sustav i općenito korištenje usluga u svim sferama života, javlja se potreba za osobnom nadogradnjom i učenjem iz razloga što, ukoliko se korisnici svih navedenih usluga ne nadograđuju i usavršavaju, postati će „informatički nepismeni“. Već smo sada svjedoci ljudi oko nas na raznim mjestima i situacijama, koji nelagodno gledajući oko sebe traže podršku za jednostavne funkcije (od podizanja novaca na bankomatu, kupnje autobusne karte na kartomatu, nadoplate bona za mobilnu telefoniju i sl.), iz razloga što nisu uhvatili korak s napretkom koji se u zadnjih 40-50 godina strahovito brzo razvija.

Osobnim stjecanjem dodatnog znanja, javlja se potreba i za programiranjem, bilo to u jednostavnom obliku ili pak profesionalnom obliku koji iziskuje visoko obrazovanje. Činjenica je da svi ti uređaju koji nas okružuju traže neko predznanje za korištenje istih, a napredovanjem, traže i naš angažman na nadogradnji svojeg znanja radi praćenja napretka i mogućnosti korištenja sve tehnologije koja nas okružuje.

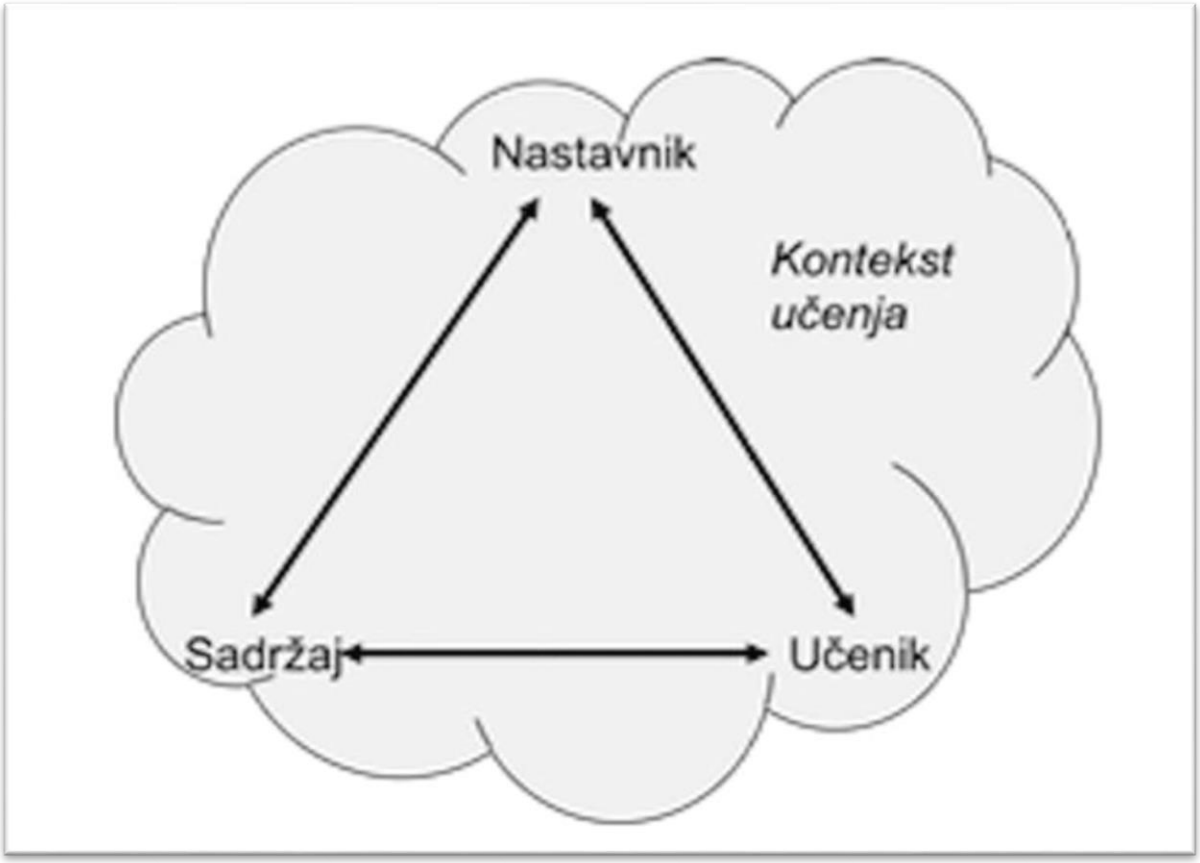
Iz tog razloga, neophodan je kvalitetan pristup i usavršavanje u svladavanju programiranja i učenja, odnosno učenja programiranja još od školske dobi pa sve do

završetka obrazovanja, neovisno od smjera životnog poziva kojeg smo odabrali, jer je predznanje i znanje u korištenju uređaja oko sebe postalo i stvar kulture osnovne naobrazbe pojedinca.

U nastavku fokusirati ćemo se na učenje i podučavanje programiranja s naglaskom na studentsku zajednicu. Studenti kao odrasle osobe koji se prvi put susreću sa programiranjem imaju raznih poteškoća pri usvajanju gradiva. Tri ključne stavke koje bi trebalo spomenuti su: slabo predznanje u rješavanju različitih logičkih problema, složenost programske sintakse te apstraktna priroda jezika. Programiranje se ne može naučiti samo kroz čitanja knjiga tj. mogu se eventualno dobiti bazični koncepti programiranja, već studenti moraju potrošiti puno vlastitog vremena kroz praktičan rad; bilo to praksa u laboratoriju ili individualan rad (samoučenje) kako bi stekli prave vještine programiranja. Dobar primjer međusobne komunikacije koja studentima pomaže u razmjeni informacija, iskustava i znanja, prikazali su u svojem istraživanju Đanić, Orehovački, Štapić (2009).

Motivacija je važan faktor kako bi studenti bili uspješniji u programiranju. Naime studenti svakako u svom obrazovanju usmjereni su prema nekom određenom cilju (npr. akademska titula, mogućnost zaposlenja i zarade, samodokazivanje ili dr.). S druge strane profesorskog gledišta najvažniji aspekt odražava se u profesorskom pristupu studentu, procjeni njegovog stupnja obrazovanja te prilagodbi i eventualnom individualnom pristupu pojedinom studentu.

Kako bi u konačnici postigli zadovoljavajući rezultat učenja, uz prethodna dva navedena uvjeta, izuzetno bitnu ulogu sačinjava i kvaliteta samog programa, odnosno, materije koja se obrađuje. Gradivo mora biti prvenstveno interesantno, prilagođeno stupnju obrazovanja studenta, profilirano na pozitivnim primjerima iz realnog života i taksativno posloženo da se ga može svladavati postupno uz slikovite primjere koji uključuju primjere vježbi i izrade primjera programiranja koji se koriste u svakodnevnoj praksi. Jedna od boljih koncepata takvog učenja ostvaruje se pomoću vizualnih alata koje ćemo spomenuti u narednog poglavlju.



Slika 13. Didaktički trokut (Krpan, 2015, str. 2)

3.1 Alati za dijagram toka

Dijagrami toka su alati koji pomažu korisniku u razumijevanju i percepciji smisla i toka procesa, odnosno programiranja. Svaki je korak sistematski prikazan različitim simbolom te se uz njega pojavljuje kratak opis postupka. Sam tijek procesa vizualno je prikazan na način da se od simbola do simbola prikazuje strelica koja određuje smjer tijeka procesa.

Takvi alati korisni su jer na relativno jednostavan način slikovito simplificira prikaz procesa, čime ga čini korisniku jednostavnijim i pristupačnijim. Grafički prikaz simbola na jednostavan način predočava tijek rješavanja problematike koja prethodi kompletiranju samog procesa.

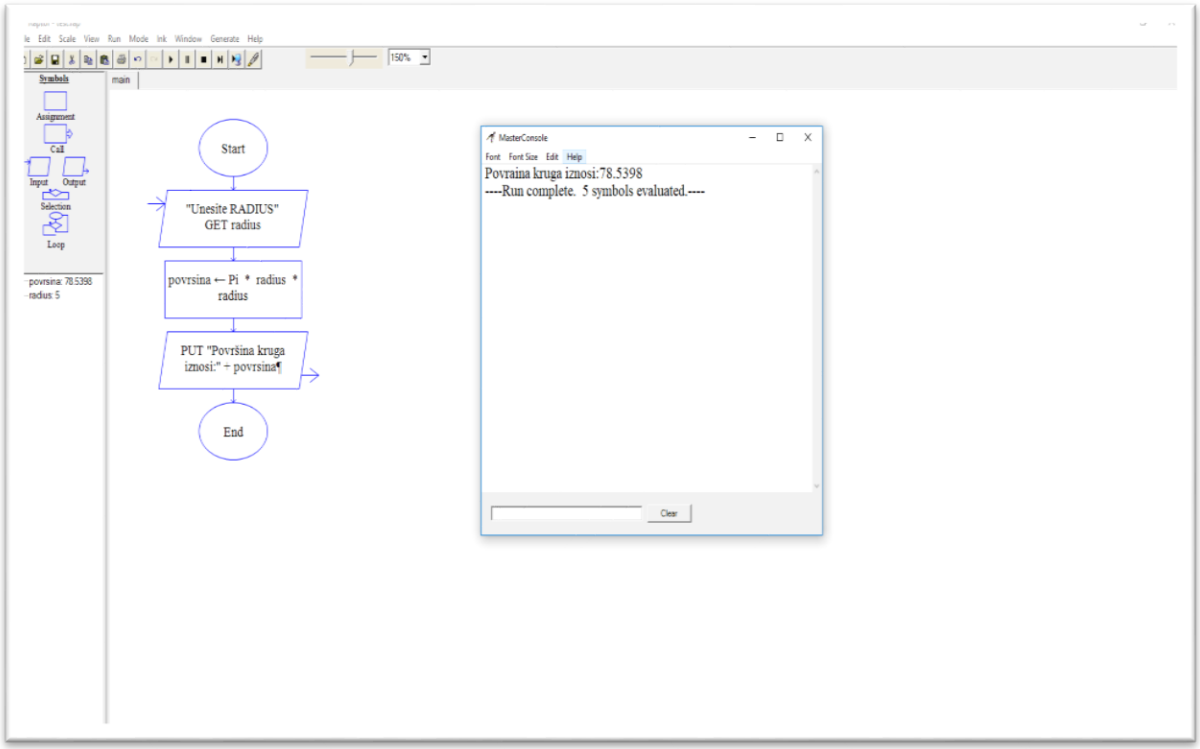
Također, takvi alati mogu znatno ubrzati rješavanje procesa, a paralelno s time povećati razumijevanje problematike, upravo radi vizualne raščlambe, pruža se korisniku šira slika cijelog procesa, naspram mogućnosti zapinjanja na pojedinoj

prepreci, ukoliko se rješava problematika na jedan od konvencionalnih načina izravnim pisanjem. U nastavku opisati ćemo na zoran način primjenu dijagrama toka.

3.1.1 RAPTOR

RAPTOR je vizualno programsko okruženje temeljeno na dijagramu toka, posebno dizajnirano kako bi studentima pomoglo da vizualiziraju algoritme i izbjegavaju sintaktičke pogreške. Programi u RAPTORU stvoreni su vizualno, a njihovo izvođenje prati slijed dijagrama. U nastavku opisati ćemo najvažnije značajke RAPTORA:

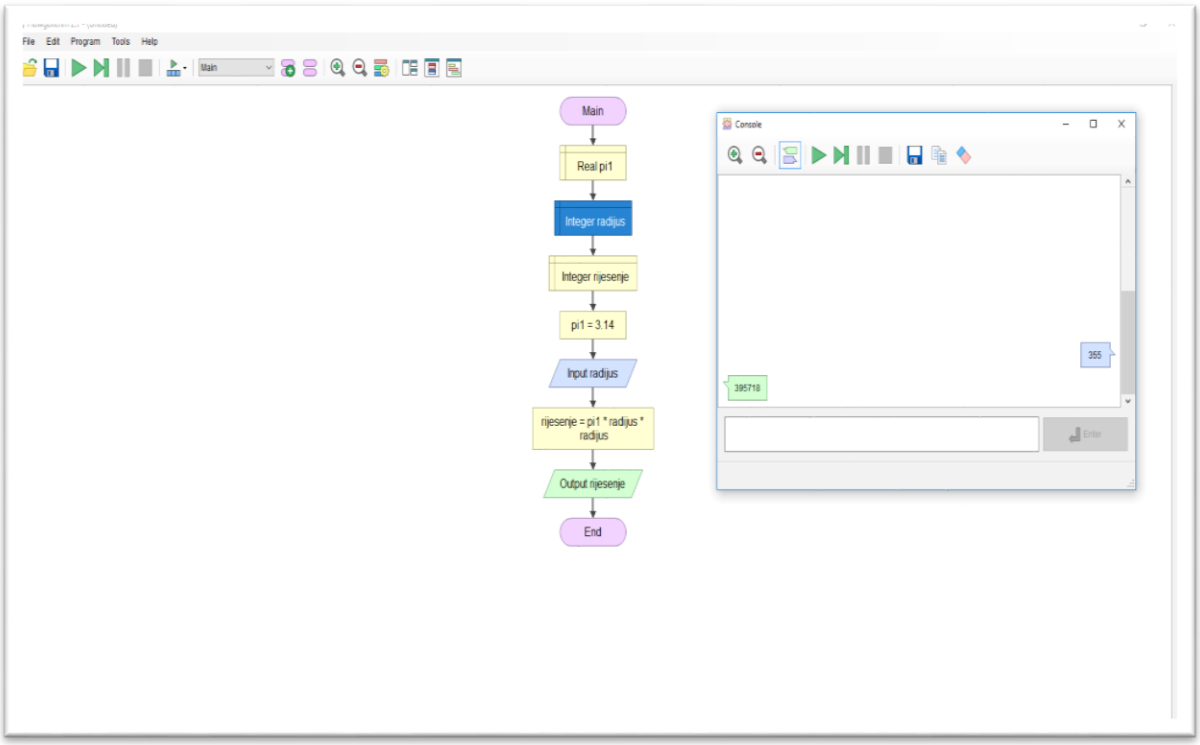
- RAPTOR razvojno okruženje smanjuje količinu sintakse koju korisnik mora naučiti za ispravno stvoriti program,
- RAPTOR razvojno okruženje je vizualno te se simboli izvršavaju jedan za drugim što donosi prednost lakšeg praćenja izvođenja programa,
- RAPTOR poruke o pogreškama su osmišljene na način da budu razumljive programerima početnicima.



Slika 14. RAPTOR - dijagram toka izračuna površine kruga

3.1.2 Flowgorithm

Flowgorithm je vizualno programsko okruženje koje korisnicima omogućava pisanje i izvršavanje programa koristeći dijagram toka. Alat je osmišljen na način da potiče razvijanje opće logike (algoritama), a ne programskog jezika. Flowgorithm ima mogućnost pretvaranja osmišljenog dijagrama toka u današnje popularne programske jezike kao što su: C++, C#, Java, Python, Visual Basic i druge. Sastoji se od standardnih simbola za početak i kraj, uvjetna grananja, ulaze, izlaze i blokova naredbi.



Slika 15. Flowgorithm - računanje površine kruga

3.2 Mini - jezici

Mini - jezici su skupina vizualnih jezika koji unutar sebe sadrže alate kojima si pomažemo u učenju programiranja.

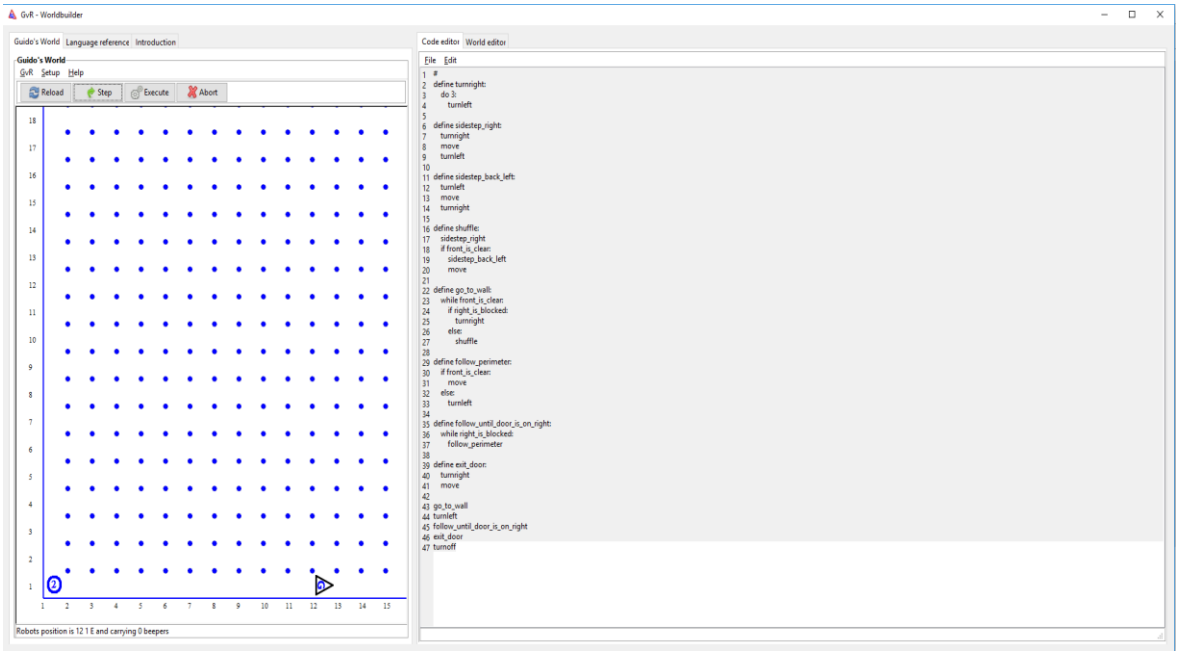
Njih se kvalitetno može iskoristiti pri učenju programiranja iz razloga što na vizualan i intuitivan, a opet vrlo jednostavan i učinkovit način, uvode studenta u svijet programiranja. Možemo ustvrditi da su mini jezici zajedno s svojim alatima, jedan od temeljnih i općih pomoćnika koji će budućeg programera uvesti u svijet programiranja, pružiti mu uvid algoritamsko razmišljanje, ali na vrlo pristupačan i zabavan način. To je jedan od bitnih faktora zbližavanja studenta s sastavom gradiva, na način da student razumije osnovne postavke projektiranja i prevlada strah od nepoznatog. Pomoću mini - jezika i pripadajućih alata, student stječe širi pogled na zadatke koji su pred njima, a profesori imaju mogućnost individualno se posvetiti studentima kroz ove mini - jezike.

Sama ideja stvaranja mini - jezika bila je aktualna još osamdesetih godina prošlog stoljeća, a prvi veći uspjeh postignut je implementacijom grafičke kornjače u mini -jezike, kao osnovnog logotipa, čijom se animacijom i grafičkim mogućnostima kornjače, sve više razvijaju alati zajedno s mini grafičkim jezicima i to sve do današnjeg dana. Tako je i danas osnovna ideja mini jezičnog pristupa programiranju, stvaranje prigodnog dizajna koji kroz mini - jezike pomaže studentima početnicima u programiranju svladati osnovne početke i olakšati prijelaz u zahtjevnije programerske korake. Osnovno načelo jest djelovanje kroz subjekt unutar programiranja (kornjača, robot ili dr.), kojim upravlja student i kroz nizove naredbi, odgovora na grupe pitanja ili sl. te tako stvara mini programe kojima subjekt vodi k cilju.

Osnovni cilj ovakvih mini - jezika jest stvaranje temeljnih znanja i vještina za kasniji prelazak na učenje i rad u višim jezicima, što je obrađeno u nastavku ovog završnog rada. Bitno je također napomenuti da se kroz mini jezike i pripadajuće alate stječe široki spektar znanja koji pomaže studentima svladavanje dijelova viših programskih jezika, a sve to na intuitivan i zabavan način.

3.2.1 Guido van Robot

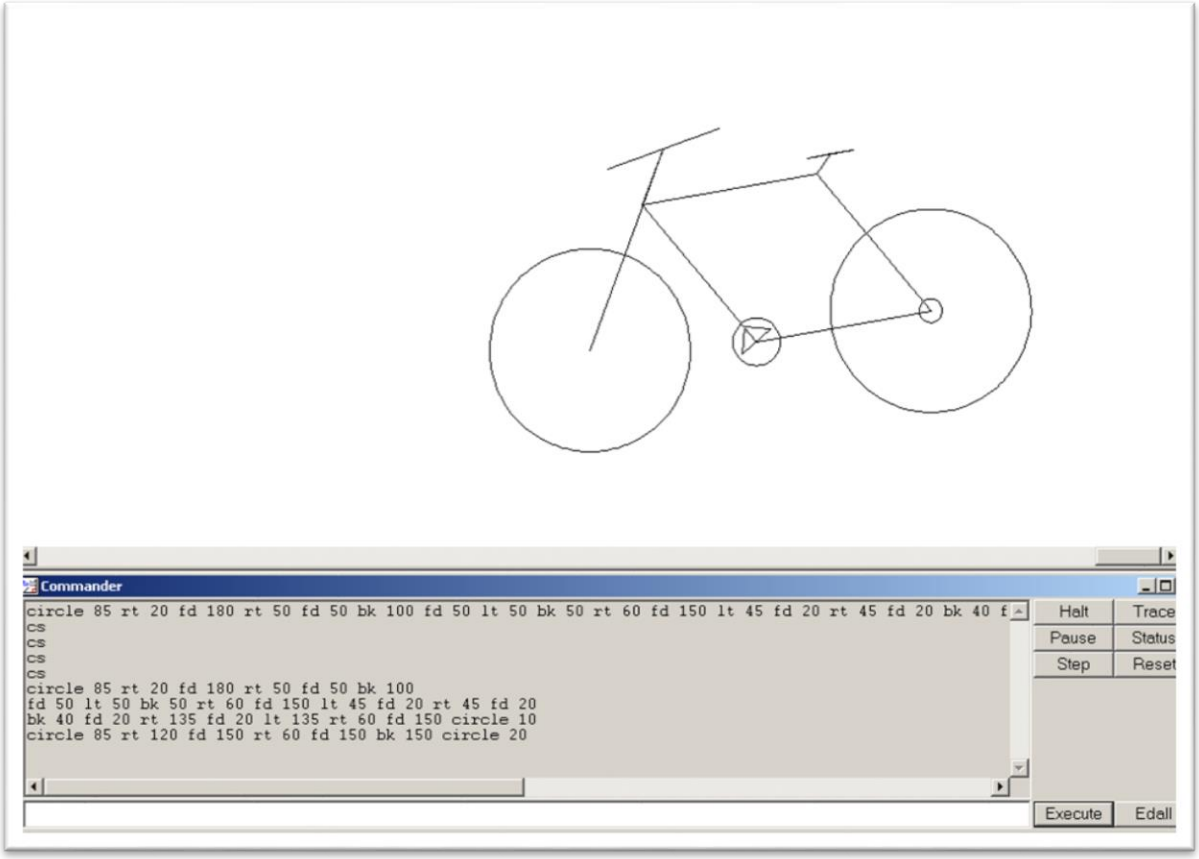
Guido van Robot je besplatan alat namijenjen učenju osnova programiranja. Bazira se na minimalističkom programskom jeziku koji pruža dovoljno sintakse korisnicima / učenicima kako bi naučili pojmove uvjetnog grananja, petlji te proceduralne apstrakcije. Najveća značajka ovog alata je učenje u programskom okruženju te rješavanje problema uz povratnu vizualnu informaciju. Ukratko, to je uvodni programski jezik za učenje osnovnih pojmova koji je kasnije primjenjiv na bilo koje jezik više razine.



Slika 16. Guido van Robot

3.2.2. MSWLogo

MSWLogo je vizualni programski jezik stvoren za obrazovnu uporabu, ponajprije za konstruktivističko učenje. Prva verzija alata nastala je 1968. godine u Cambridgeu. Glavna ideja alata je omogućiti crtanje po ekranu dajući naredbe robotu (kornjači), pri čemu pisanjem naredbi korisnici trebaju voditi kornjaču kuda treba ići kako bi nacrtali željeni oblik. Alat je dostupan za Windows i Mac platformu.



Slika 17. MSWLogo

3.3 Algoritmi za vizualizaciju

Još jedan od alata koji spada u skupinu vizualnih alata za sortiranje, omogućuje svladavanje početnih problema i strahova te koji uvelike pomaže studentima početnicima u programiranju, jest alat za vizualizaciju algoritama ili skraćeno ViSA.

3.3.1 ViSa

Kako su u svom radu zorno prikazali Reif i Orehovački (2012), spomenuti alat ViSA pomaže studentima u vizualizaciji algoritma sortiranja. Njegova odlika jest relativno jednostavno postavljanje i kompletan automatski sustav za vizualizaciju s prikazanim objašnjenjima, ali i usporedbom algoritma sortiranja.

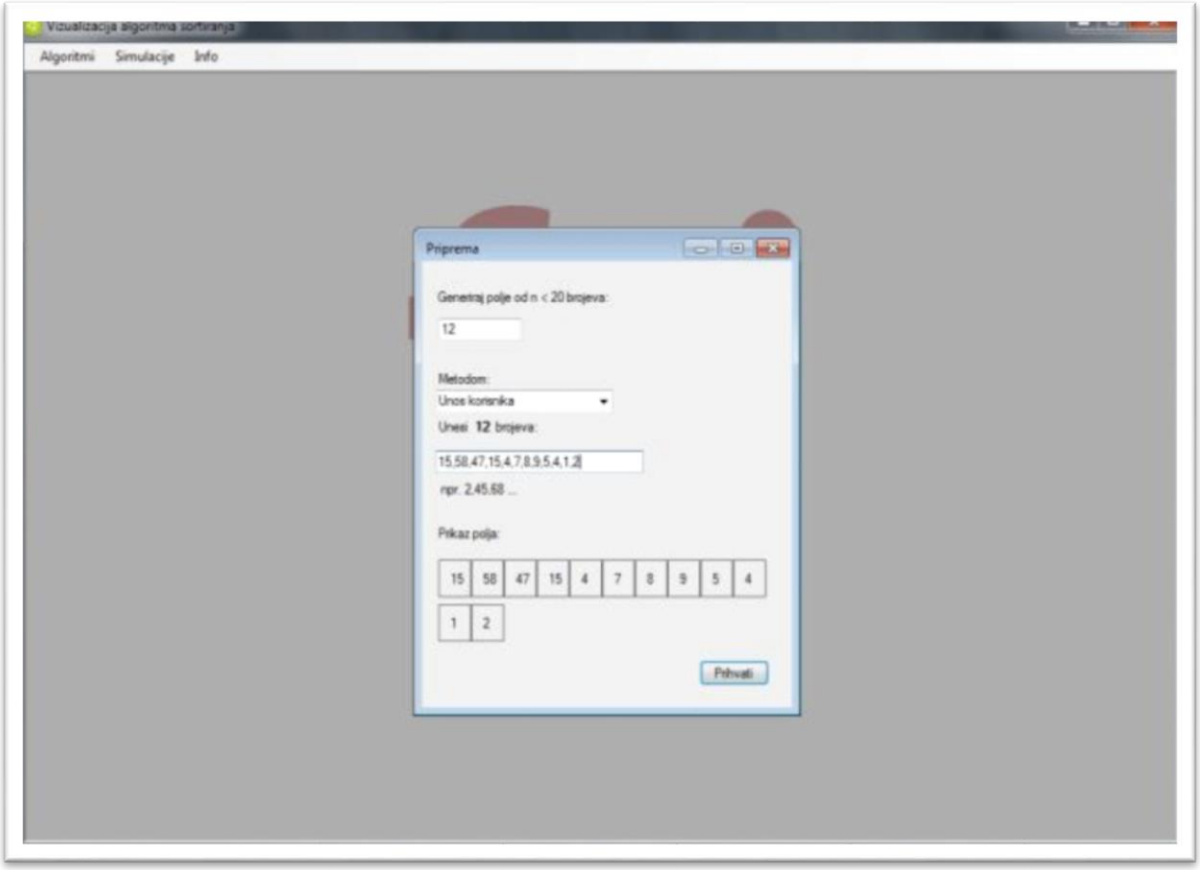
Slijedom navedenog, bitno je napomenuti da je alat ViSA vrlo kvalitetan alat koji pruža mogućnost jednostavnog korištenja, kroz vrlo jednostavno sučelje, koje pruža niz povratnih informacija i jednostavan je za učenje. Bitno kod takve vrste alata jest brzina integracije, brzi pristup jednim klikom kojim je alat odmah spreman za uporabu. On mora omogućiti korisniku konstrukciju vlastitih skupova ulaznih podataka te mora omogućiti istovremeno vizualizacijsku interakciju.

Unutar alata ViSA moguće je raditi s najpoznatijim algoritmima sortiranja, kao što su Cocktail Sort, Comb Sort, Heap Sort, Insertion Sort, Merge Sort, Quick Sort, Selection Sort i Shell Sort. Nakon odabira jednog od algoritama, korisnik bira obrazac, u kojem definira tekstualni okvir, pregled polja, oznake i dr. Takvi su elementi dinamički te se bilo kada mogu izvršiti pojedine izmjene koje neće utjecati na primitak ispravnog unosa podataka.

Ukoliko je korisnik učinio pogrešku, od strane alata dobiti će poruku o pogrešno unesenim ili pak pogrešno formuliranim podacima. Također, korisnik može izabrati jednu od metoda generiranja slučajnih brojeva kao što su čiste slučajne vrijednosti, velike vrijednosti, male vrijednosti.

Ako pak korisnik unese određenu vrijednost, to čini putem praznih tekstualnih polja, što za rezultat ima pokretanje generiranja polja. Nakon što korisnik dovrši s implementacijom svih skupova podataka, pokreće se konačni test unesenih podataka te na taj način omogućuje korisniku nastavak programiranja. Na ovaj se način omogućuje korisniku da segmentalno provjerava svoj rad, kako bi u konačnici bio

siguran da će dijelovi algoritama ispravno funkcionirati prilikom animacije na kraju svog rada.



Slika 18. ViSA - alat vizualizacije aloritama sortiranja

4. Skupina viših jezika za programiranje zajedno s alatima za pomoć učenju programiranja

Nakon uspješnog svladavanja opće logike programiranja kroz vizualizacijske alate, kao što smo u prethodnom dijelu završnog rada naveli, u nastavku učenja programiranja prelazi se na izbor prvog jezika za programiranje.

Prilikom izbora prvog jezika za programiranje, važno je definirati značajke koje budući izabrani jezik mora zadovoljiti. Kriteriji koje jezik mora zadovoljavati u što većem postotku jesu:

- da li je jezik primjeren općem okviru zajednice studenata i kao takav pogodan za kvalitetno praćenje nastave,
- da li jezik odgovara modernom pristupu prenošenja znanja u učenju programiranja,
- da li jezik podrazumijeva kao logičan slijed učenja rješavanje problema individualnim pristupom i podrškom studentima na način da pruža fleksibilno razvojno okruženje sukladno potrebama studenata,
- da li jezik posjeduje kvalitetnu literaturu i nastavne materijale te da li je moguće koristiti stečeno znanje i nakon završenog obrazovanja.

U nastavku ćemo navesti i opisati nekoliko jezika i alata koji zadovoljavaju iznad navedene kriterije i kao takvi pogodni su za logičan prelazak stjecanja znanja sa vizualizacijskih alata za programiranje na svladavanje znanja i vještina programiranja viših programskih jezika.

Prilikom svakog važnog koraka, postoji strah od novog i bojazan od neuspjeha pa tako kod studenata koji započinju s programiranjem u nekom od svjetskih jezika dolazi do smanjenja samopouzdanja. U ovom važnom koraku prelaska na „pravo“ programiranje, izuzetno je važna komponenta prilagodbe programa studentima, podrška koju dobivaju od profesora, individualan pristup u što većoj mjeri te u konačnici alati koji će biti od velike pomoću studentima programerima. Prvenstveno se to odnosi na alate koji pomažu kod analize koda.

Alati za analizu koda pomažu programeru analiziranje putem raznih sugestija, pronalaženja anomalija, pisanja pravilne sintakse koda, utvrđivanja grešaka unutar

koda i sl. Upravo ovakve, naizgled male pomoći, izuzetno su važne za programere početnike, jer ih uz te sugestije i pomoći pomažu dovesti do cilja, a u važnim prijelomnim trenutcima potiču studente da ne odustanu, već im malim pomoćima i poticajima usmjeravaju u nastavak izvršavanja zadataka. A upravo dolazak do cilja i uspješno izvršavanje zadataka, osobna su satisfakcija svakog programera. Kad povučemo paralelu s počecima programiranja kroz igrice, tada možemo reći da je uspješno izvršen zadatak unutar programiranja, osjećaj jednak onome kada smo pobjeđivali u igricama. Jer u konačnici djetinjstvo i sjećanja su uvijek jedan sastavni dio nas samih, kojih se s veseljem prisjetimo.

4.1 C++ i Verifikator

4.1.1 C++

C++ je opće namijenjen objektno orijentirani jezik koji se izvorno zvao "C s klasama". U programiranju C++ poznat je kao snažan jezik koji dozvoljava kontroliranje memorije, brzinu te efikasnost koda.

U nastavku nabrojati ćemo značajke C++ jezika:

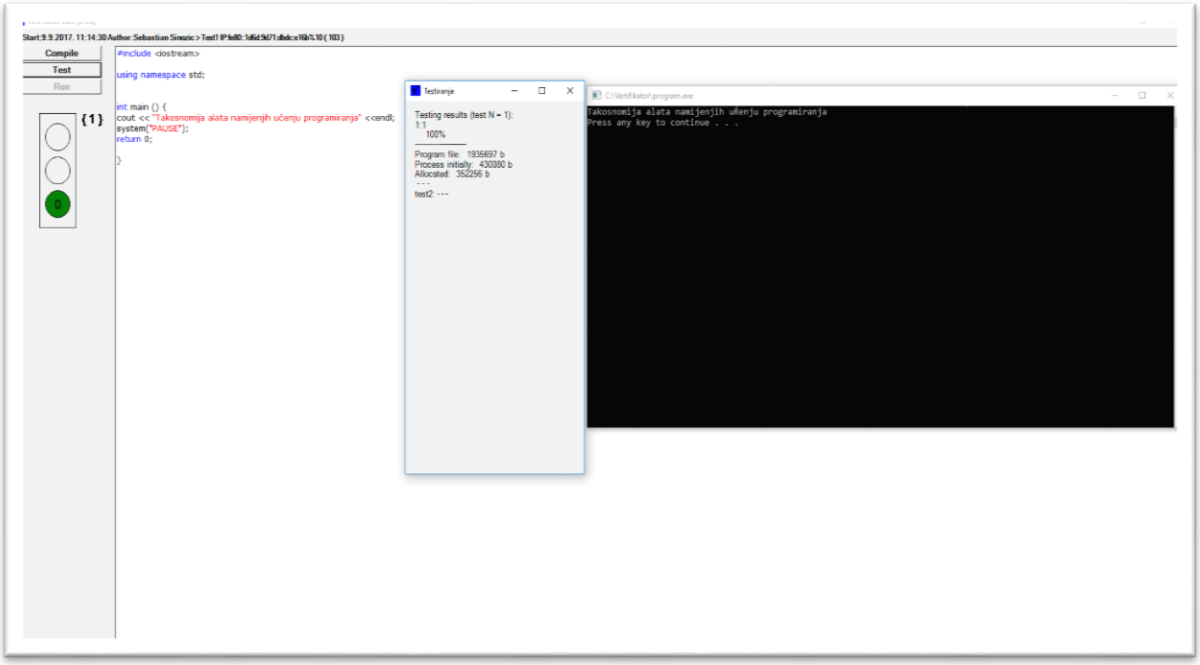
- C++ je prenosiv jezik (eng. portable language) te je često izbor za razvoj većeg broja uređaja i raznih platformi,
- C++ ima bogat izbor već gotovih biblioteka,
- C++ dopušta upravljanje iznimkama i preopterećenje funkcijama koje nisu podržane u nekim jezicima,
- C++ koristi se za širok raspon aplikacija – od aplikacija do 3D grafike za igre pa sve do matematičkih simulacija u realnom vremenu.

4.1.2 Verifikator

Nastavno na navedeno, opisati ćemo alat Verifikator kojeg su autori analizirali u radu Radošević, Orehovački, Lovrenčić (2009), a koji pomaže učenju programiranja u C++ okruženju. Verifikator je alat namijenjen pomaganju studentima radi lakšeg stjecanja programskih vještina i poboljšanja procesa odlučivanja.

Verifikator radi s Dev-C++ razvojnim okruženjem, a ima sljedeće značajke:

- Mogućnost personalizacije programa – Studenti unose vlastite podatke (ime, prezime, broj indeksa i druge.) koji se zapisuju u programski kod u obliku komentara kako bi se kasnije moglo provjeriti da li je program pisan u verifikatoru.
- Vremensko ograničenje – Studenti u zadanom vremenskom roku moraju izvršiti zadani zadatak. Nakon isteka vremena program se može pokretati, ali se ne može mijenjati.
- Autentičnost koda – Verifikator sprječava unošenje koda iz vanjskih izvora.
- Usvajanje pravilnih programskih navika – Verifikator zahtjeva provjeru koda svakih deset novih unosa koji su prikazani na semaforu u grafičkom obliku. Nakon deset unosa verifikator ne dopušta pokretanje koda.



Slika 19. Verifikator - alat koji pomaže pri učenju programiranja

4.2 Java i QAPlug

4.2.1 Java

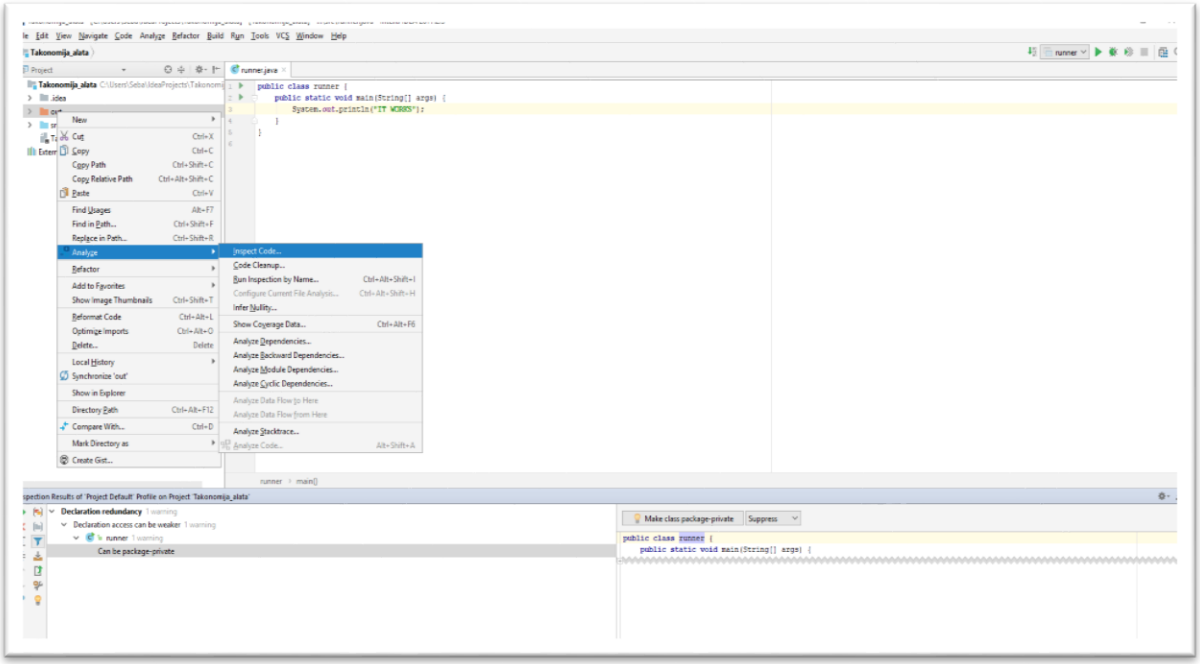
Java je viša razina objektno-orijentiranih jezika stvorena od strane Sun Microsystems. Dizajnirana je na sličan način kao C++, ali sa jednostavnijom sintaksom. Java je jezik opće namjene koja sadrži niz značajki, a one su prvenstveno orijentirane za korištenje na okruženjima kao što je www - svjetska mreža (eng. www - World Wide Web). U nastavku ćemo nabrojati i opisati neke od glavnih značajki Jave:

- Lakoća korištenja: Temelji jave bazirani su na programskom jeziku C++. On je vrlo moćan jezik sa kompleksnom sintaksom te nije adekvatan za određene potrebe Jave. Iz tog razloga Java je poboljšala neke ideje C++ jezika na način da bude također vrlo moćan ali u isto vrijeme i jednostavniji.
- Sigurnost: Pošto je Java usmjerena na mobilne uređaje koji izmjenjuju podatke putem mreža, odlikuje se vrlo visokom sigurnošću kako bi se spriječilo bilo kakvo curenje podataka prilikom korištenja.
- Neovisnost platforme: Programi moraju funkcionirati ispravno bez obzira na kojoj su platformi pokrenuti. Java je napravljena da bude prenosiv jezik (eng. portable language), odnosno, da nesmetano radi na bilo kojoj platformi.

4.2.2 QAPlug

U nastavku opisati ćemo alat QAPlug. Naime QAPlug je besplatan alat koji se može dodati u integrirano programersko okruženje Jave te pomaže upravljanjem kvalitete koda. Korisniku omogućuje odabir različitih funkcionalnosti kao što su:

- PMD (statički analizator koda) koji pronalazi neiskorištene varijable, nepotrebne objekte koji su stvoreni i dr.
- CheckStyle (analizator koda) koji provjerava više aspekata koda. Pronalazi probleme oko dizajna samih klasa, probleme s dizajnom metoda te također provjerava izgled i probleme s oblikovanjem koda.
- FindBugs (analizator koda) koji pronalazi potencijalne greške u kodu (eng. Bug) ili neispravnosti u radu.



Slika 20. QAPlug - alat za detaljnu analizu koda

4.3 Phyton i Prospector

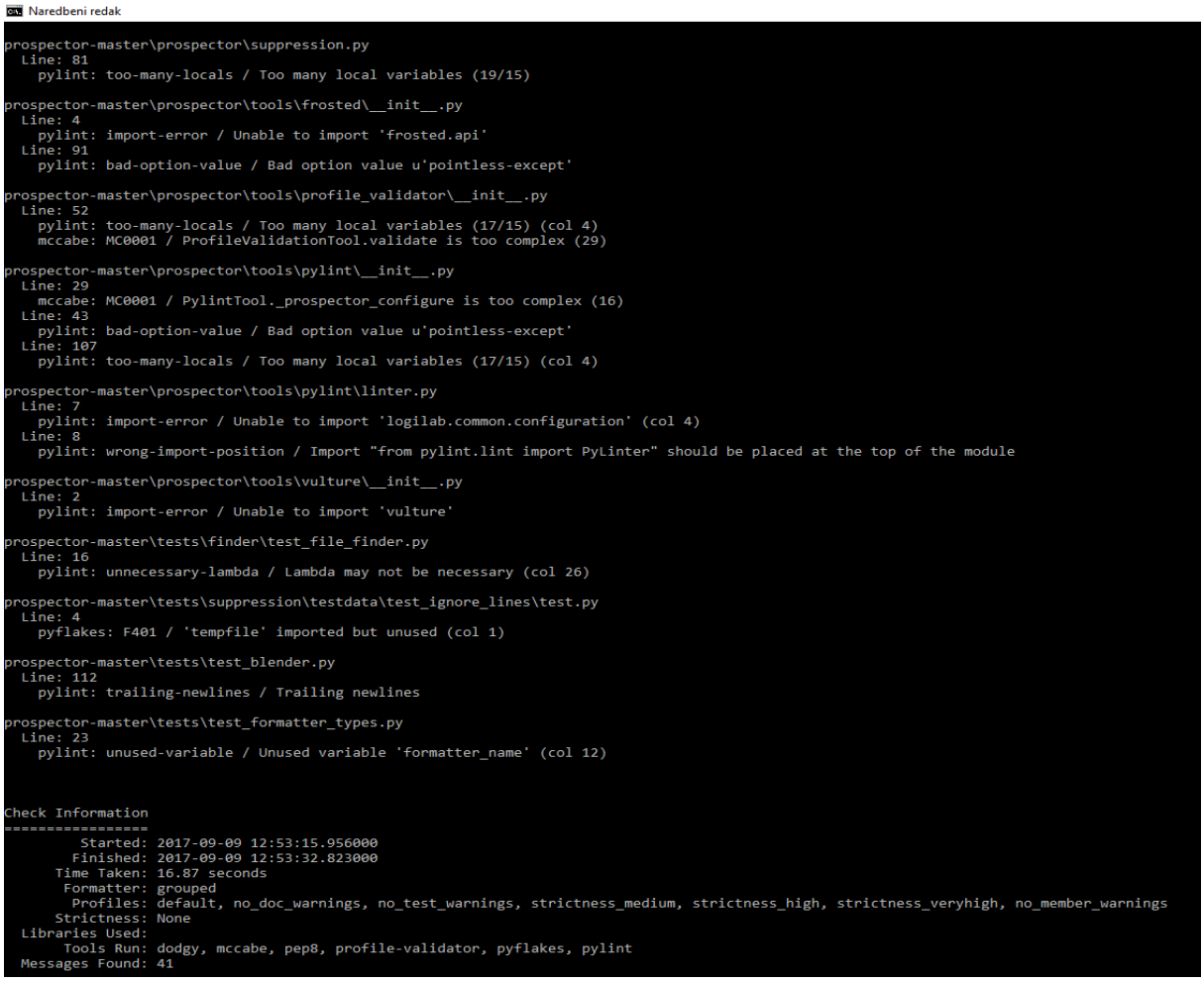
4.3.1 Python

Python je objektno orijentiran programski jezik s dinamičnom semantikom, stvoren 1991. godine od strane Guida van Rossum-a. Python koristi jednostavnu sintaksu koja je lagana za čitati i time smanjuje troškove održavanja samog programa. Također jezik podržava module i pakete, što potiče modularnost i ponovnu uporabu koda. U nastavku opisati ćemo glavne značajke Python jezika:

- Jednostavnost učenja – Python koristi nekoliko ključnih riječi, sadrži jednostavnu strukturu i jasno definiranu sintaksu što omogućuje studentima brzo učenje jezika.
- Jednostavan za čitanje – Phytonov kod je jasno definiran i jasno uočljiv
- Prenosivost – Python podržava veliki broj platformi te sadrži isto sučelje na svim platformama
- Proširivost – Python sadržava mogućnost dodavanja modula. Moduli omogućuju programerima dodavanje ili prilagođavanje alata za što bolju učinkovitost.

4.3.2 Prospector

Prospector je alat za statičku analizu koda, razvijen od strane LandScape, koji pruža izlazne informacije o pogreškama, potencijalnim problemima i drugo. Za razliku od drugih alata za analizu koda kojima je potrebno duže vrijeme da bi se prilagodili osobnim stilovima programiranja, Prospector nudi nekoliko osnovnih profila koji su već predisponirani za pružanje određenih izlaza uz odabrane knjižnice koje se nalaze u projektima.



Slika 21. Prospector - analizator Python koda

5. Usporedba alata namijenjenih učenju programiranja

Skupina alata:	Vrsta alata:	Ime alata:	Kriteriji		
			Motiviranost korisnika	Svladavanje straha od programiranja	Smanjivanje odustajanja od studiranja
Prva skupina: Alati za učenje programiranja u okruženju Web 2.0	Najpogodniji alati i aplikacije za učenje programiranja u Web okruženju	Clou9 IDE	***	****	****
		Ideone	***	****	****
		JDoodle	***	****	****
		Repl.it	*****	****	****
	Blog i wiki kao kolaborativni alati za učenje programiranja	Blog	*****	****	*****
		wiki	*****	****	*****
Druga skupina: Alati za učenje programiranja putem video igara	Igre za učenje programiranja namijenjene djeci	Tynker	****	****	****
		Waterbear	****	****	****
		RoboMind	****	*****	*****
		Kodable	****	****	****
	Igre za učenje programiranja namijenjene studentima	CodeCombat	****	*****	*****
		Code Hero	*****	*****	*****
		Code Hunt	****	****	***
Treća skupina: Vizualni alati za učenje programiranja	Alati za dijagram toka	Raptor	*****	*****	****
		Flowgorithm	****	****	***
	Mini - jezici	Guido van Robot	****	****	****
		MSWLogo	****	****	****
	Algoritmi za vizualizaciju	ViSA	*****	****	****

OCIJENE: [*] – LOŠE, [**] – DOVOLJNO, [***] – DOBRO, [****] – VRLO DOBRO, [*****] – ODLIČNO

Skupina alata:	Vrsta alata:	Ime alata:	Kriteriji		
			Motiviranost korisnika	Svladavanje straha od programiranja	Smanjivanje odustajanja od studiranja
Četvrta skupina: viši jezici za programiranje zajedno s alatima za pomoć učenju programiranja	Viši programski jezici	C++ i Verifikator	*****	*****	*****
		Java i QAPlug	*****	****	****
		Phyton i Prospector	****	***	****

Tablica 1. Usporedba alata namijenjenih učenju programiranja

OCIJENE: [⋆] – LOŠE, [⋆⋆] – DOVOLJNO, [⋆⋆⋆] – DOBRO, [⋆⋆⋆⋆] – VRLO DOBRO, [⋆⋆⋆⋆⋆] – ODLIČNO

Sukladno usporedbi prikazanoj u prethodnoj tablici, korisniku se na temelju navedenih alata preporučuju slijedeći alati, podijeljeni po skupinama:

- Unutar prve skupine - alati za učenje programiranja u okruženju Web 2.0, podskupina alata i aplikacija za učenje programiranja u Web okruženju, najviše se ističe alat Repl.it, prvenstveno radi zasebne mogućnosti koje alat pruža profesoru, za stvaranje „virtualne učionice“, u kojoj studentima može zadavati zadatke te dodatno sugerirati i prezentirati pojašnjenja.

  U drugoj podskupini - blog i wiki kao kolaborativni alati za učenje programiranja, nakon analize i prikaza u ovom radu, ipak se wiki za nijansu ističe kao prigodniji alat za učenje u programiranju, prvenstveno radi svoje široke primjene i interaktivne suradnje većeg broja korisnika, mogućnosti kolaboracije te raspoloživosti alata za učenje.
- Unutar druge skupine - alati za učenje programiranja putem video igara, najviše od svih igara ističe se igra Code Hero. To je od svih navedenih igara jedina igrica u kojoj je postavljena mogućnost izmjene i reorganizacije samog sadržaja igrice, ovisno o afinitetima i željama igrača, a sve kroz izmjenu koda.
- Unutar treće skupine - Vizualni alati za učenje programiranja, najpogodniji od svih obrađenih jesu Raptor i Guido van Robot. Raptor je od svih prikazanih alata najjednostavniji za korištenje, vrlo je pristupačan i logičan te ima veliku

podršku na raznim internetskim izvorima te je najzastupljeniji među studentskom zajednicom. Od mini – jezika, Guido van Robot na logičniji način obrađuje linije koda te pruža veći spektar mogućnosti i funkcionalnosti (npr. petlja).

- Unutar četvrte skupine - viši jezici za programiranje zajedno s alatima za pomoć učenju programiranja, spominje se jezik Python, koji je trenutno najzastupljeniji jezik na svijetu. Unatoč tomu, preporučuje se jezik C++ i Verifikator, i to ne radi samog jezika, već prvenstveno radi alata Verifikator, koji pruža puno veće mogućnosti naspram ostalih navedenih alata s pojedinim jezicima, prikazanim u ovom radu, kao što su; nameće periodičnu kontrolu koda (svakih 10 redaka), ne dopušta ubacivanje kodova iz vanjskih izvora te osim klasičnog upozorenja o pogrešci, pruža sugestivniji uvid u mjesto pogreške s pripadajućim opisom i pojašnjenjem, a sve u svrhu ispravke učinjene pogreške.

Zaključno, potrebno je napomenuti da su svi alati obrađeni u ovom završnom radu dobri i preporučljivi studenskoj zajednici te ostalim korisnicima, međutim, u iznad navedenom tekstu, pojašnjeni su najistaknutiji alati sistematizirani po skupinama, sve sukladno sadržaju završnog rada.

Zaključak

Napretkom sveopće tehnologije i industrije, koja u zadnjih par desetljeća napreduje izuzetno brzim tempom, dovodi do stalnog pojavljivanja novih, suvremenijih i sve više kompleksnih uređaja, koji imaju potrebu da se njima upravlja. Jedan dio tih uređaja podložan je izravnom upravljanju i izravnom korištenju, dok je dobar dio tih uređaja predisponiran da se njima upravlja putem određenih programa ili pak u kombinaciji s izravnim i programiranim upravljanjem.

Iz tog razloga, potreba za znanjima i vještinama iz programiranja nikada nije bila na višem stupnju. Doslovce, za svaki aspekt našeg života potrebno je nešto programirati, od najmanjih korisnih stvari pa do složenih programskih cjelina. Da bi se mogli nositi s takvom brzinom napredovanja i da bi mogli koristiti tehnologiju i uređaje u obimu koji nam oni pružaju, nužno je kontinuirano samoučenje i rada na usavršavanju već stečenih znanja i vještina. Ako pak usmjerimo pogled u budućnost, pretpostavka je da će se razvoj tehnologija sve više razvijati, a time će potreba za znanjima i vještinama programiranja sve više rasti, baš kao i potreba za stvaranjem stručnjaka koji će nam takva znanja i vještine moći pružiti na kompleksnijim uređajima, odnosno potrebama.

Da je tomu tako, svjedoči i nedavna odluka Vlade Republike Hrvatske kroz dio kurikularne reforme donijela odluku kojom se određuje da se Informatika kao redovni nastavni predmet mora interpolirati u razrednu nastavu već od slijedeće školske godine. Takvom odlukom, Vlada RH stvorila je pretpostavku za što raniju mogućnost interakcije djece u školstvu s informatičkim okruženjem, bez obzira na njihov socijalni status, čime se omogućuje stjecanje znanja i vještina od najranije učeničke dobi, odnosno, sukladno ovom završnom radu, što ranije upućivanje budućih korisnika u skupine alata primjerene njihovom interesu kroz obrađene i prikazane skupine, kao osnovne smjernice.

Zasigurno će se na takav način iz svih tih interesnih učeničkih skupina u konačnosti jedan dio opredijeliti za nastavak informatičkog obrazovanja, koji će ih u konačnosti kroz proces visokog obrazovanja dovesti i do zvanja programera, a kao što smo u uvodu naveli, imati će posla na pretek.

Ovaj završni rad može prema svemu navedenom, uz potencijalne buduće programere, koristiti i roditeljima, kao odraslim osobama, kako za pobliže

upoznavanje s ovom tematikom, tako i za osobno usavršavanje, a svakako kao jednu od polaznih točaka za prepoznavanje djetetovih afiniteta, ali i za stjecanje predznanja o sigurnosti korištenja alata za programiranje koji će njihova djeca koristiti.

Svjesni smo da svakodnevno napreduju i alati za programiranje, kao i svi ostali alati i aplikacije koje u svakodnevnoj uporabi koristimo. Međutim, sukus i forma prikazana kroz ovaj završni rad ne mijenjaju se u odnosu na (pre)brzi napredak tehnologije, već nas kao korisnike, odnosno programere početnike i uznapredovale programere definiraju kroz duže vremensko razdoblje.

Iz tog razloga, ovaj završni rad postaje vrlo svrsishodna literatura za širi krug čitatelja, kako bi se blagovremeno opredijelili za svoje interese, čime mogu uštedjeti mnogo vremena i nepotrebnog znanstvenog lutanja, ali možda i promijeniti svoje interese ka jednoj drugoj skupini alata, koja ih nakon čitanja više zanima od inicijale zamisli prije početka čitanja.

Literatura

1. BLINKLIST, *Ideone Review: Testing Your Code*, 2013
   http://blinklist.com/reviews/ideone
   (pristupljeno: 28.kolovoza.2017)

2. BLOODSHEDSOFTWARE, Dev-C++: Providing Free Software to the internet community, 2015
   http://www.bloodshed.net/devcpp.html
   (pristupljeno: 28.kolovoza.2017)

3. BMC Med Educ, A new generation of web-based tools for virtual collaborative education, 2006
   https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1564136/
   (pristupljeno: 07.kolovoza.2017)

4. Chokshi, M. Eclipse, *NetBeans or IntelliJ IDEA – Which Is The Best IDE For Java Development?*, 2017
   https://www.linkedin.com/pulse/eclipse-netbeans-intellij-idea-which-best-ide-java-mrunal-chokshi
   (pristupljeno: 30.kolovoza.2017)

5. Chudy,L., DZONE Sonar and QAPlug integration, 2011
   http://www.bloodshed.net/devcpp.html
   (pristupljeno: 28.kolovoza.2017)

6. Codeacademy*, Web 1.0 i Web 2.0 - Razlike, 2013*
   https://sites.google.com/site/nikolinacrnkoviccode/pocetna-stranica/web-1-0-i-web-2-0
   (pristupljeno: 28.kolovoza.2017)

7. Code Hunt, *What is code hunt?*, 2014
   https://www.codehunt.com/about.aspx

(pristupljeno: 21.kolovoza.2017)

8. CODE HERO, *Code Hero: Hiding a text book in a video game*, 2012
   http://www.ign.com/articles/2012/03/01/code-hero-hiding-a-text-book-in-a-video-game
   (pristupljeno: 22.kolovoza.2017)

9. FINNLY, HOW TO PROGRAM USING MSW LOGO, 2009
   http://www.instructables.com/id/How-to-use-MSW-Logo/
   (pristupljeno: 02.rujna.2017)

10. GPML, *Teaching tools*, 2017
    http://lana.foi.hr/laboratory/index.php?id=teaching-tools
    (pristupljeno: 20.kolovoza.2017)

11. Đanić, M.; Orehovački, T.; Stapić, Z. (2009) *Introducing CaCM: toward new students collaboration model,* Zagreb: University of Zagreb
    (preuzeto: kolovoz 2017.)

12. Khara, A. *How good is the website CodeCombat.com at teaching JavaScript, Python, and its associated languages?*, 2015
    https://www.quora.com/How-good-is-the-website-CodeCombat-com-at-teaching-JavaScript-Python-and-its-associated-languages
    (pristupljeno: 25.kolovoza.2017)

13. James, M., Waterbear – a visual language for Javascript, 2011
    http://www.i-programmer.info/news/98-languages/2389-waterbear-a-visual-language-for-javacript.html
    (pristupljeno: 30.kolovoza.2017)

14. JDOODLE, Easy and Quick way to compile and run C++ Code Online, 2017
    https://www.jdoodle.com/online-compiler-c++
    (pristupljeno: 28.kolovoza.2017)

15. Krpan, D. (2013) *Poučavanje programera početnika u visokom obrazovanju*, Split: Prirodoslovno-matematički fakultet Split
(pristupljeno: 27. kolovoza 2017.)

16. Kurdi, S., Teach children basic programming concepts, with RoboMind, 2013
http://www.freewaregenius.com/teach-children-basic-programming-concepts-with-robomind/
(pristupljeno: 30.kolovoza.2017)

17. Minayev, A. Blog Basics, What Is a Blog?, 2011
https://blogbasics.com/what-is-a-blog/
(pristupljeno: 06.kolozovza.2017)

18. MISSCNEWTON, App Review – Coding Apps: Kodable, 2013
https://primaryipadclassroom.com/2013/06/22/app-review-coding-apps-kodable/
(pristupljeno: 30.kolovoza.2017)

19. Perez, S., Tynker A startup that teaches kids to code, now work with robots, drones and more, 2015
https://techcrunch.com/2015/05/15/tynker-a-startup-that-teaches-kids-to-code-now-works-with-robots-drones-and-more/
(pristupljeno: 28.kolovoza.2017)

20. RAZI, Raptor flowchart quick start for beginners, 2013
https://primaryipadclassroom.com/2013/06/22/app-review-coding-apps-kodable/
(pristupljeno: 02.rujna.2017)

21. Orehovački, T.; Babić, S. (2015) *Inspecting Quality of Games Designed for Learning Programming*, Zagreb: University of Zagreb
(preuzeto: kolovoz 2017.)

22. O'REILLY, T. *What is web 2.0*, Radar, 2009.

23. Radošević, D.; Orehovački, T.; Lovrenčić, A. (2009) *New Approaches and Tools in Teaching Programming*, Zagreb: University of Zagreb
(preuzeto: kolovoz 2017.)

24. Radošević, D.; Orehovački, T.; Lovrenčić, A. (2009) *Verificator: Educational Tool for Learning Programming*, Zagreb: Faculty of Organization and Informatics
(preuzeto: kolovoz 2017.)

25. Reif, I.; Orehovački, T (2012) *ViSA: Visualization of Sorting Algorithms*, Varaždin: Faculty of Organization and Informatics
(preuzeto: kolovoz 2017.)

26. Revolvy, Flowgorithm, 2014
https://www.revolvy.com/main/index.php?s=Flowgorithm&item_type=topic
(pristupljeno: 02.rujna.2017)

27. Stanislao, L., *Ready, steady, program! How children can learn coding (and teach numeracy to a robot) during STEM school visit events, 2015*
http://bejlt.brookes.ac.uk/paper/ready-steady-program-how-children-can-learn-coding-and-teach-numeracy-to-a-robot-during-stem-school-visit-events/
(pristupljeno: 03.rujna.2017)

28. Škorić, I.; Pein, B.; Orehovački, T. (2016) *Selecting the Most Appropriate Web IDE for Learnigng Programming Using AHP*, Pula: Juraj Dobrila University of Pula
(preuzeto: kolovoz 2017.)

29. 0X7DF*, Python code analysis using Prospector, 2015*
http://www.bloodshed.net/devcpp.html
(pristupljeno: 28.kolovoza.2017)

Popis slika

Slika 1. Razlike između Web 1.0 i Web 2.0 8
Slika 2. Clou9 IDE okruženje ..... **Error! Bookmark not defined.**
Slika 3. Ideone online kompilator 11
Slika 4. JDoodle online kompilator 12
Slika 5. Repl.it online programsko okruženje 13
Slika 6. Tynker - igra za najmlađe 19
Slika 7. Waterbear igra spajanja blokova 20
Slika 8. RoboMind igra pomicanja tenka 21
Slika 9. Kodable igra za djecu 22
Slika 10. CodeCombat igra pomoću Python sintakse 23
Slika 11. Code Hero igra unutar igre 24
Slika 12. CodeHunt igra programske logike 25
Slika 13. Didaktički trokut 28
Slika 14. RAPTOR - dijagram toka izračuna površine kruga 29
Slika 15. Flowgorithm - računanje površine kruga 30
Slika 16. Guido van Robot 32
Slika 17. MSWLogo 33
Slika 18. ViSA - alat vizualizacije aloritama sortiranja 35
Slika 19. Verifikator - alat koji pomaže pri učenju programiranja 38
Slika 20. QAPlug - alat za detaljnu analizu koda 40
Slika 21. Prospector - analizator Python koda 41

Popis tablica

Tablica 1. Usporedba alata namijenjenih učenju programiranja 43

Sažetak

Prilikom izrade taksonomije alata za učenje programiranja, potrebno je pronaći veliki broj podataka koji nam mogu biti korisni za provedbu same klasifikacije i stvaranja skupina alata za korištenje prilikom učenja programiranja, i to prvenstveno prema namjeni koja se namjerava koristiti za učenje.

Radi specifičnosti teme i pronalaska relevantne količine podataka, potrebno je posjetiti i analizirati veći broj internetskih stranica, obraditi ih prema tematskoj smjernici i analizirati, kako bi se mogle stvoriti osnovne skupine kao što je u radu prikazano.

Obzirom da smo se opredijelili da izvor podataka i smjernica koji će nam pomoći u stvaranju taksonomije budu izvori s raznih internetskih stranica (foruma, te drugih kolaboracijskih izvora koji su neobično velikog značaja), samo početak istraživanja prilikom izrade ovog rada bio je usmjeren ka internet izvorima.

Prema samom naslovu teme, na Internetu ima jako malo izravnih podataka pa je potrebno proširiti fokus traženih ulaznih podataka kao parametara neophodnih za izradu same taksonomije. U interakciji s raznim stranicama, uočen je izuzetno velik broj komentara, tema ili cjelina, koji spominju nesnalaženje korisnika alata ili pak korištenje krivih alata za svoje interese, te je iz tog razloga potreba za ovakvom taksonomijom alata za učenje programiranja tim viša.

Nakon dovršetka izrade analize prikupljenih podataka te klasifikacije alata, kompletna tematska cjelina čini se još više korisna nego što je zapravo na početku govorio sam naslov teme. Nakon početnog problema oko iznalaženja izvora za analizu, sve većim angažmanom i pronalaženjem podataka, ukupnim rezultatima ovaj završni rad još više nam govori o značaju taksonomije za sve korisnike pametnih uređaja i tehnike, bez obzira na njihov stupanj znanja i socijalni status, odnosno da pruža univerzalan uvid svim zainteresiranim čitateljima.

Važne riječi: taksonomija, Internet, alati, programiranje

Abstract

When designing the taxonomy of the programming learning tools, it is necessary to find a large number of data that can be useful for implementing the classification itself and creating a set of tools to use when learning programming, primarily for the purpose to be used while learning.

Due to the specific topic and finding a relevant amount of data, it's necessary to visit and analyze a major number of web – sites, process them according to the thematic guidelines and analyze them, so that basic groups can be created as shown in the paper.

Given that the determination of the direction of the data and guidelines that would enable the creation of taxonomy was the source of various internet sites (forums and other collaborative sources of extraordinary importance), even the very beginning of the research in this work was directed at internet sources.

According to the subject's headline, there are very few direct data on the Internet, so the focus of the input data needs to be expanded as parameters necessary for taxonomy. In interaction with various sites, a remarkably large number of comments, themes, or entities have been noted that mention the user's lack of tools or the use of the wrong tools for their own interests, and for this reason the need for such a taxonomy of the programming learning tool is higher.

After completing the analysis of the data collected and the classification of tools, the complete thematic set seems to be even more useful than it was at the beginning of the title theme. After the initial problem of finding the source for the analysis, with increasing engagement and finding of data, the overall results of this final work are even more telling about the importance of taxonomy for all users of smart devices and techniques, regardless of their degree of knowledge or knowledge and social status provides a universal insight to all interested readers.

Keywords: taxonomy, Internet, tools, programming