

Usporedna analiza alata za učenje programiranja

Miličević, Marijela

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:021321>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-18**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli

Fakultet ekonomije i turizma

"Dr. Mijo Mirković"

MARIJELA MILIČEVIĆ

**USPOREDNA ANALIZA ALATA ZA UČENJE
PROGRAMIRANJA**

Završni rad

Pula, 2020.

Sveučilište Jurja Dobrile u Puli

Fakultet ekonomije i turizma

"Dr. Mijo Mirković"

MARIJELA MILIČEVIĆ

USPOREDNA ANALIZA ALATA ZA UČENJE PROGRAMIRANJA

Završni rad

JMBAG: 0303059458, redovita studentica

Studijski smjer: Poslovna informatika

Predmet: Osnove programiranja

Znanstveno područje: Društvene znanosti

Znanstveno polje: Ekonomija

Znanstvena grana: Poslovna ekonomija

Mentor: izv.prof.dr.sc. Tihomir Orehovački

Pula, 2020.



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Marijela Miličević, kandidat za prvostupnika ekonomije/poslovne ekonomije, smjera poslovna informatika, ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega 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, Marijela Miličević dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom "*Usporedna analiza alata za učenje programiranja*" 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, _____ godine.

Potpis

Sadržaj

1. Uvod	1
2. Općenito o alatima za učenje programiranja	3
3. Posebne skupine alata za učenje programiranja	5
4. Kriteriji za izradu usporedne analize alata za učenje programiranja	6
5. Web alati i suradničko učenje programiranja	8
5.1. Web 2.0 alati za učenje programiranja	9
5.1.1. <i>PaizaCloud IDE</i>	10
5.1.2. <i>Codeanywhere</i>	11
5.1.3 <i>Usporedna analiza Web 2.0 alata za učenje</i>	12
5.2. Suradničko učenje pomoću društvenih Web aplikacija	13
5.2.1. <i>Blog</i>	14
5.2.2. <i>Wiki</i>	15
5.2.3. <i>Usporedna analiza alata za suradničko učenje na Web-u</i>	16
6. Vizualni alati za učenje programiranja	17
6.1. Alati za učenje kroz dijagram toka	18
6.1.1. <i>RAPTOR</i>	19
6.1.2. <i>FlowGorithm</i>	20
6.1.3. <i>Gliffy</i>	21
6.1.4. <i>Usporedna analiza alata za izradu dijagrama toka</i>	22
6.2. Alati namijenjeni vizualizaciji algoritma	24
6.2.1. <i>ViSA</i>	25
6.2.2. <i>PITON</i>	26
6.2.3. <i>Usporedna analiza alata namijenjenih vizualizaciji algoritma</i>	27
6.3. Mini-jezik kao alat za učenje programiranja	28
6.3.1. <i>MSWLogo</i>	28
6.3.2. <i>Guido van Robot</i>	30
6.3.3. <i>Usporedna analiza mini-jezika</i>	31
7. Igre za učenje programiranja	32
7.1. Igre za djecu	33
7.1.1. <i>Daisy the dinosaur</i>	33
7.1.2. <i>Algorithm city: Coding game</i>	34
7.1.3. <i>Usporedna analiza igri za djecu</i>	35

7.2. Igre za studente.....	36
7.2.1. CodeCombat.....	36
7.2.2. Screeps.....	38
7.2.3. Usporedna analiza igri za studente	39
7.3. Alati za stvaranje i uređivanje igri	40
7.3.1. GameMaker-Studio 1.4	40
7.3.2. Construct 2.....	41
7.3.3. Usporedna analiza alata za stvaranje i uređivanje igri	43
8. Učenje programiranja u višim programskim jezicima	44
9. Učenje programiranja u višim programskim jezicima opće namjene	46
9.1. Programski jezik Python	46
9.1.1. Prospector.....	48
9.1.2. Pylint	49
9.1.3. Usporedna analiza alata programskog jezika Python	50
9.2. Programski jezik Java.....	51
9.2.1. QAPlug.....	52
9.2.2. Checkstyle.....	53
9.2.3. Usporedna analiza alata programskog jezika Java	54
9.3. Programski jezik C++	55
9.3.1. Verificator	57
9.3.2. Cppcheck	59
9.3.3. Usporedna analiza alata programskog jezika C++	60
10. Učenje programiranja u višim programskim jezicima za rad na Web-u	61
10.1. PHP	61
10.1.1. Zend Studio.....	63
10.1.2. Rapid PHP Editor	64
10.1.2. Usporedna analiza alata za PHP	65
10.2. JavaScript.....	66
10.2.1. Atom.....	67
10.2.2. Sublime Text	68
10.2.3. Usporedna analiza alata za JavaScript	69
11. Zaključne usporedne analize alata za učenje programiranja	70
11.1. Web 2.0 okruženje zaključna analiza	71

11.2. Vizualni alati zaključna analiza	72
11.3. Igre za učenje programiranja zaključna analiza.....	73
11.4. Viši programski jezici zaključna analiza.....	74
12. Zaključak	76
13. Literatura	78
14. Popis slika	83
15. Popis tablica	84
Sažetak.....	85
Abstract	86

1. Uvod

Rad na računalu danas se smatra jednim od najiskorištenijih znanja i vrlo je važan kako za djecu tako i za odrasle. U većini školskih sustava diljem Europe, ali i svijeta općenito, rad na računalu je jedno od najvažnijih karakteristika, a samim time ono uvelike pomaže kod zapošljavanja ljudi u različitim sektorima i različitim kadrovima. Godinama unazad u Hrvatskoj je vrlo mali dio školskom kurikulumu bio posvećen upravo programiranju na nastavi informatike i samim time je i nastala motivacija za ovaj završni rad.

Razvoj logičkog razmišljanja nekim ljudima je i dalje nepoznanica, no ono kao takvo uistinu krase uspješne poslovne ljude diljem svijeta. Upravo takav način logičkog razmišljanja potiče djecu od malih nogu da probleme razvrstavaju na dijelove i samim time dolaze do ciljeva na zanimljivije i učinkovitije načine.

Samim time, odnosno vježbom i rješavanjem laganih, a potom i onih težih zadataka djeca stvaraju logiku u svom načinu razmišljanja što je zapravo neophodno kod učenja programiranja. Upornost i velika posvećenost te zainteresiranost i motivacija za one naizgled nebitne i vrlo male detalje, stvaranje logike i vježbanjem razmišljanja te povezanost u timove najbolje su kombinacije za stvaranje dobrih programa i dalje učenje programiranja. Programiranje kao jedna od grana informatike bitno je u napredovanju svih velikih tehnologija i sveopće populacije.

Postoji veliki broj alata koji su namijenjeni upravo olakšanom pristupu i lakšem učenju programiranja. U ovom završnom radu fokus je na četiri skupine, odnosno četiri različita načina primjene alata kako bi se djeci, učenicima, studentima, ali i ljudima koji nisu usko vezani za informacijsko-komunikacijske tehnologije olakšalo učenje programiranja, a to su:

- Web 2.0 alati za programiranje te Web 2.0 suradničko učenje programiranja
- Vizualni alati za učenje programiranja
- Alati za programiranje kroz igre i to u kategorijama igri za djecu, studente te alate za stvaranje i uređivanje igri
- Alati podrške za učenje programiranja u višim programskim jezicima

U prvoj skupini tj. kategoriji nalaze se alati koji su namijenjeni učenju programiranja koristeći sučelje Web 2.0. Podijeljeni su u dvije podskupine kao jedni od popularnih alata, te blog i wiki kao podskupina suradničkog učenja programiranja.

U drugoj skupini nalaze se vizualizacijski alati za učenje programiranja. Kao takvi, služe upravo jednoj vrsti prelaska programiranja iz faze u fazu, odnosno programiranja igri pa sve do viših programskih jezika. Upravo je na taj način i napravljena usporedba alata koji su od pomoći korisnicima kako bi zapravo ušli u ozbiljno učenje programiranja. Također, pod ovu kategoriju se svrstavaju i mini-jezici koji su kroz usporedbu omogućili detaljnu sliku i pomoć ka ulasku u programiranje u višim programskim jezicima.

Kako je već spomenuto, treća skupina je namijenjena učenju programiranja kroz igru. Kategorije su podijeljene na igre za djecu, tj. igre koje su namijenjene mlađim uzrastima pa sve do kategorija igri za studente i na kraju kategorija grafičkog programiranja igri koja je danas uvelike popularna u svijetu. Kroz samu igru lako je zadobiti veliku zainteresiranost korisnika jer je to jedna od najboljih vrsti zabave, ali i učenja u istom trenutku što uvelike pomaže da se djeca, studenti, ali i ljudi koji nisu dio informacijsko-komunikacije tehnologije zainteresiraju za programiranje.

Posljednja i najvažnija kategorija je ona sa alatima koji su namijenjeni upravo učenju programiranja u višim programskim jezicima. Zajednički cilj skoro svim studentima i korisnicima je naučiti znanja i vještine kroz najpopularnije programske jezike što ih jednog dana dovodi do akademskih obrazovanih građana, a samim time oni kroz rad i trud prerastaju u ozbiljne programere i odlične IT stručnjake.

Takva vrsta kategorizacije i usporedbe napravljena je upravo kako bi se korisnik odmah na početku mogao opredijeliti za alate koji mu najviše odgovaraju i zadovoljavaju kriterije, a ne kako bi nepotrebno gubio vrijeme na nešto za što će se nakon nekog vremena saznati kako uopće nije potrebno te kako bi se učenjem programiranja započelo upravo od samih osnova i dobilo pomoć od alata koji su upravo tome i namijenjeni.

2. Općenito o alatima za učenje programiranja

Učenje programiranja danas zahtjeva vrlo složen proces od strane nastavnika u školama i profesora na fakultetima. Prvenstveno, učenike i studente treba motivirati jer ih jako puno dolazi sa različitim predznanjima, odnosno dolaze ili samouka ili samo kroz spominjanje programiranja na nastavi informatike. Strah od programiranja danas je kod mladih izrazito velik, motivacija im nestaje nakon prvog problema u pisanju koda, no u novije vrijeme dostupan je sve veći broj alata koji su razvijeni upravo u svrhu educiranja učenika, studenata, no dostupni su i mnogobrojni internet izvori kako bi se i ostalim korisnicima omogućilo obogaćenje znanja.

Upravo zbog toga postoji veliki broj pristupa programiranju i veliki broj alata koji su razvijeni kako bi se pomoglo i olakšalo razumijevanje programskog koncepta. Uz to, postoji velik broj literature za različite programske jezike, nebrojeno veliki broj video predavanja i različitih video uradaka na temu olakšavanja pisanja kodova na internetu i u programskim jezicima.

Generalno, alati za učenje programiranja su posebna sučelja namijenjena za učenje. Takvi alati za učenje mogu se podijeliti u nekoliko osnovnih skupina i to kao alati koji su kreirani za razvoj softvera ili posebno kreirani alati za edukaciju. Alate se može podijeliti u četiri osnovne skupine:

1. Alati namijenjeni programiranju koristeći Web 2.0 i suradničko učenje programiranja
2. Vizualizacijski alati namijenjeni učenju programiranja kao što su posebni alati namijenjeni vizualizaciji, alati namijenjeni dijagramu toka, ali i mini-jezici
3. Učenje programiranja kroz igre kao alat za učenje programiranja se pokazalo dosta motivirajuće prema mlađim generacijama, ali i samim studentima koji su se odlučili za studij informatike
4. Jedni od najpopularnijih i najrazvijenijih alata danas se koriste u višim programskim jezicima koji su danas najpopularniji za programiranje

Već samoj djeci koja pokažu želju i zainteresiranost za učenjem programiranja omogućeno je kroz različite alate, igre ili mini-jezike da svoje znanje upotpunjuju od samog početka i samim time prate razvoj različitih programskih jezika, različitih namjena i težina. Nadalje, studentima alati omogućuju korištenje različitih mogućnosti u različitim programskim jezicima kako bi im se učenje programiranja kao takvog olakšalo.

Krajnji rezultat najčešće nije ono što je u startu trebao biti, nego se mijenja ako dođe do nekakvog problema. Programiranje se ni u kojem smislu nije uvijek vezalo na samo pisanje koda jer postoji mnogo različitih načina, različitih alata kako se dolazi do samog programa kojeg će određena skupina ljudi koristiti. Razvoj nekakvog programa, aplikacije ili softvera je dugotrajan proces koji obuhvaća jako veliki broj istraživanja, analiziranja i prije svega razmišljanja i težnje ka cilju. Čitljivosti koda, razumna logika i sama sintakse su ključni zadaci kako bi sam kod jednog dana zapravo postao program ili u slučaju razvoja nekog softvera postao zapravo sam softver pogodan za korištenje većoj skupini ljudi.

Pojavom nove inačice, tj. pojavom Web 2.0. tehnologije i uvođenjem HTML-a, napredak u programiranju u programima poput C#, C++, JavaScript-a omogućio je još više dodatnih načina za razvoj programa, što je definitivno dovelo do novijih alata namijenjenih učenju programiranja u svim dobima, sve od djece pa do razvojnih programera koji se poslom programiranja bave godinama.

3. Posebne skupine alata za učenje programiranja

Kroz povijest literature vidljivo je kako je predloženo mnoštvo različitih taksonomija.

Prema Pears-u et al. (2007, 204-223) posebne skupine alata za učenje programiranja dijele se na:

- **Vizualizacijski alati** – kroz različite vizualne primjere prikazuju neke od određenih dijelova izvršenja samog programa. Omogućuju korisnicima, uglavnom djeci da kroz vizualizaciju nekog programskog rješenja (na primjer crtanja nekog geometrijskog tijela) vide kroz koju vrstu koda i naredbi dolazi do idejnog rješenja
- **Alati za automatsko ocjenjivanje** – omogućuju korisnicima (najčešće studentima) da svoje završene programe predaju samom alatu te im u tom slučaju i sam alat dodjeljuje ocjenu. Takvi alati se koriste najčešće u unaprijed pripremljenim slučajevima kojima se automatski provjerava radi li program koji je predan ispravno. U većini slučajeva upravo te ocjene koje sam alat daje omogućuju studentima da vide gdje točno griješe u pisanju koda i kako bi sami svoje znanje usavršili i naučili na greškama.
- **Alati za podršku u programiranju** – oni najčešće koriste sve alate koji su nabrojani u ovim posebnim skupinama, ali i još dodatni broj njih. Dakle, sadrže različite vrste programskih okolina koje su dostupne i namijenjene u edukacijske svrhe. Sadrže i vizualizacijske alate (ako je riječ o nekom projektu koji zahtjeva na primjer nekakav crtež) kako bi što lakše dočarao stvarnu percepciju.
- **Alati programske okoline** – ovakva vrsta alata posebno je prilagođena i namijenjena upravo za edukacijske svrhe te su to različiti oblici integriranja različitih razvojnih sučelja

Međutim, kako se i samo integrirano razvojno okruženje vremenom mijenjalo tako se i danas mijenjaju sami alati za učenje programiranja. Web je stvoren kako bi ljudi mogli međusobno komunicirati, slati različite vrste podataka, surađivati neovisno na kojem kraju svijeta se nalaze, mogu dijeliti svoje misli, ideje i projekte s drugim ljudima te je samim time Web 2.0 postao jedna od glavnih platformi koje i djecu i studente zanimaju za razvoj njihovih programa, ideja i na kraju velikih projekta kojeg će na kraju možda koristiti veliki broj ljudi.

4. Kriteriji za izradu usporedne analize alata za učenje programiranja

Usporedba sama po sebi znači uspoređivanje nečega s nečim na temelju nekih zajedničkih osobina. Tako se u ovom radu uspoređuju različiti alati koji služe učenju programiranja. Podijeljeni su u kategorije, no kako bi se analiza uspješno napravila potrebno je zadati određene kriterije po kojima će se analiza izvršavati.

Kriteriji za analize:

1. Zainteresiranost učenika/studenata za učenjem programiranja
2. Jednostavnost učenja
3. Dostupnost literature, materijala i vodiča
4. Pružanje informacija o pogreškama i potencijalnim problemima
5. Pružanje različitih mogućnosti i funkcionalnosti
6. Motiviranost korisnika za danjim učenjem programiranja

Zainteresiranost učenika i studenata za samim programiranjem je jedan od najvažnijih kriterija ove analize jer već u samom početku daje detalje o tome kakav stav korisnici na samom početku imaju o određenom alatu, neovisno o tome gdje su i na koji način saznali za određeni alat.

Jednostavnost učenja i dostupnost literature, materijala i vodiča su također bitni kriteriji koji vjerojatno svakog tko se sreće sa učenjem programiranja zanimaju. Postoje alati s kojima je jednostavno raditi, no postoje i oni koji iziskuju dodatan trud i rad, a dostupnost literature, materijala i vodiča su bitne stavke koje je dobro imati na umu kad se kreće u rad s programiranjem, kako u edukaciji tako i za buduće programere koji su odlučili programske jezike učiti sami.

Pružanje informacija o pogreškama i potencijalnim problemima kao i pružanje različitih mogućnosti i funkcionalnosti su definitivno jedni od bitnijih kriterija kod odabira alata koji će učenicima ili studentima pomoći u učenju jer su za to prvenstveno i namijenjeni. Svaki alat ima svoje prednosti i mane i upravo tome usporedna analiza različitih alata u različitim okruženjima i programskim jezicima služi, kako bi svatko od zainteresiranih našao po nešto za svoje potrebe i preferencije.

I za kraj, motiviranost korisnika za danjim učenjem programiranja je generalno zadnji kriterij jer upravo on daje uvid u sve pređene kriterije i zbraja sve dobre i loše strane određenih alata kako bi se stvorio završni uvid u to koliko zapravo ovakvi alati pomažu učenicima i studentima kako bi nastavili učiti i ne odustajati lako od programiranja.

Zadovoljavanje kriterija se vrši prema istraživanju literature i internet izvora, a ono može biti loše, dovoljno, dobro, vrlo dobro ili odlično. Svaka ocjena, označena je različitom bojom kao što je vidljivo u tablici 1. Nalazi se na kraju svake skupine alata, a zaključna analiza pokazuje finalne zaključke svake analize.

LOŠE
DOVOLJNO
DOBRO
VRLO DOBRO
ODLIČNO

Tablica 1 - Ocjene kriterija (*Rad autorice*)

5. Web alati i suradničko učenje programiranja

Govorili se o programiranju na WEB sučelju uglavnom se govori o inačici Web 2.0. Web 2.0. novi je standard koji se nikad nije odnosio i ne odnosi se samo na nekakva poboljšanja u tehničkom smislu nego kao socijalna i društvena platforma.

Ono što je Web 2.0 donio je to da čitatelji i sami korisnici postaju stvaratelji sadržaja objavljenog na internetu te razmjenjuju i prenose informacije direktno. Samim time zapravo i dolazi do promjena u inačicama. Dakle, zbog velike potražnje i velikog pritiska od strane tržišta poboljšao se i sustav samog interneta i njegovog korištenja.

Prema O'Reilly-u i Battlle-u (2009, 22) Web 2.0 se može podijeliti na četiri kategorije, odnosno četiri razine aplikacija:

1. **Nulta razina** – pripadaju joj aplikacije koje imaju jednaku funkciju u radu i na samoj mreži i izvan nje.
Najbolji primjer: Google Maps
2. **Prva razina** – pripadaju joj aplikacije koje imaju sklop funkcionalnosti u načinu rada van mreže, ali su iznimno važne i njihove značajke u radu koji se odvija na mreži.
Najbolji primjer: Office 365
3. **Druga razina** – pripadaju joj aplikacije čija funkcionalnost im je zaista minimalna u radu izvan mreže jer svoje najveće funkcionalnosti i sam rad održavaju na mreži.
Najbolji primjeri: društvene mreže (Facebook, Twitter, Instagram)
4. **Treća razina** – pripadaju joj aplikacije koje postoje samo na internetu i njihova je glavna zadaća u principu da bi imale što veći broj korisnika
Najbolji primjeri: Skype, Discord, e-Bay

U samom kontekstu učenja programiranja često je korišteno mnogo tehnika učenja pa tako i kolaborativno, odnosno suradničko učenje programiranja.

Kako tvrdi Dillenbourd (1999) kolaborativno (suradničko) učenje je situacija u kojoj dvoje ili više ljudi uče ili pokušavaju učiti zajedno.

Problemi koji nastaju u učenju programiranja i samom nedostatku iskusnih programera stvaraju veliku potrebu da se njihovo školovanje samim time učini što efikasnijim i efektivnijim. U samom tom kontekstu učenja programiranja najviše su korištene izokrenuta nastava (*eng. flipped classroom*) i programiranje uživo (*eng. live coding*).

- **Izokrenuta nastava** – je proces učenja kod kojeg studenti umjesto da dolaze na predavanja na fakultete oni ista ta predavanja gledaju u obliku video oblika nastave
- **Programiranje uživo** – način je kod kojeg predavač (nastavnik ili profesor) direktno piše kod pred svojim studentima i cijelo vrijeme ga ispravlja, komentira sam kod i prepravlja greške pred studentima te im objašnjava strategije kako najbolje doći do ispravnosti koda

5.1. Web 2.0 alati za učenje programiranja

Kao što je već prije rečeno, dolaskom nove inačice, odnosno nove verzije Web 2.0 u svijet tehnologija omogućila se velika i novija razina komunikacije među korisnicima na internetu. Kako se komunikacija razvijala između korisnika i računala ona je omogućila da se sad komunikacija izražava između na primjer dva korisnika međusobno i samim time je dovela korisnike na vrlo visoku poziciju te im omogućila aktivno korištenje. Korisnici su postali aktivni sudionici u samom procesu stvaranja web sadržaja i dovela ih do novih razina društvenog i socijalnog.

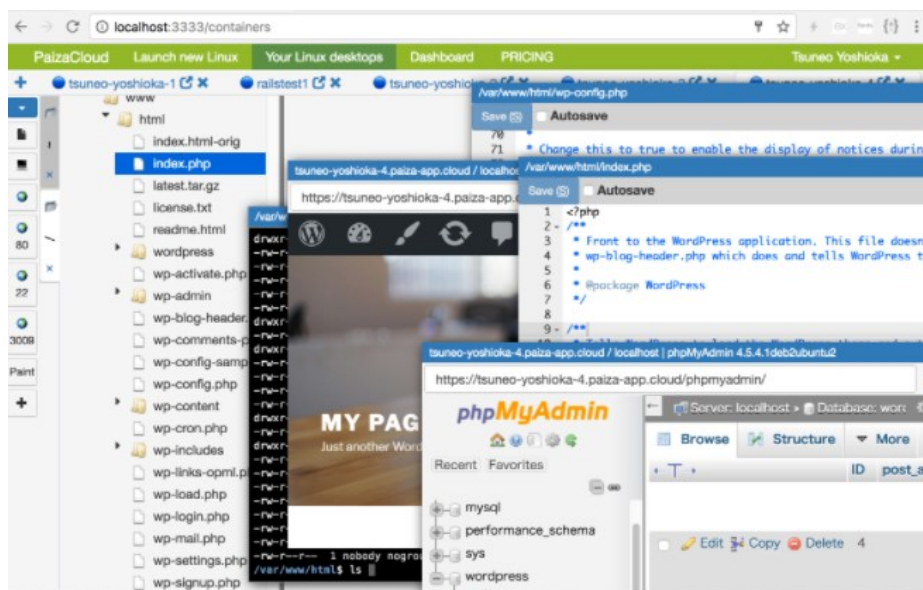
Postoji dakako određeni broj razlika između okruženja u Web 2.0 i Web 1.0, no jedno od najvažnijih i najzastupljenijih je upravo socijalizacija više korisnika na jednom mjestu. Dvostruka komunikacija omogućila je sudjelovanje u stvaranju sadržaja i distribucije informacija, društvenih umrežavanja, spajanja raznih informacijskih sadržaja sa velikog broja izvora i direktna razmjena informacijskih sadržaja.

5.1.1. PaizaCloud IDE

PaizaCloud IDE je integrirano razvojno okruženje koje podržava veliki broj programskih jezika kao što su C, C++, PHP, JavaScript, Python, Ruby i dr. Napisan je gotovo kompletno u programskom jeziku JavaScript dok se u pozadini rada nalazi Node.js.

Obzirom da se radi o programiranju na Web-u, ono programerima automatski daje mogućnost da započnu kodiranje na temelju svog preglednika kao što je prikazano u slici 2.

Među najboljim karakteristikama se ističu ugrađeni terminali sa naredbama, dovršavanje koda za različite identifikatore, pomoć oko samog uređivanja, a samim time i istovremeno i višestruko uređivanje. Također koristi upravljač datotekama, uređivač samog teksta te različite podrške za dodavanje drugih dodataka.



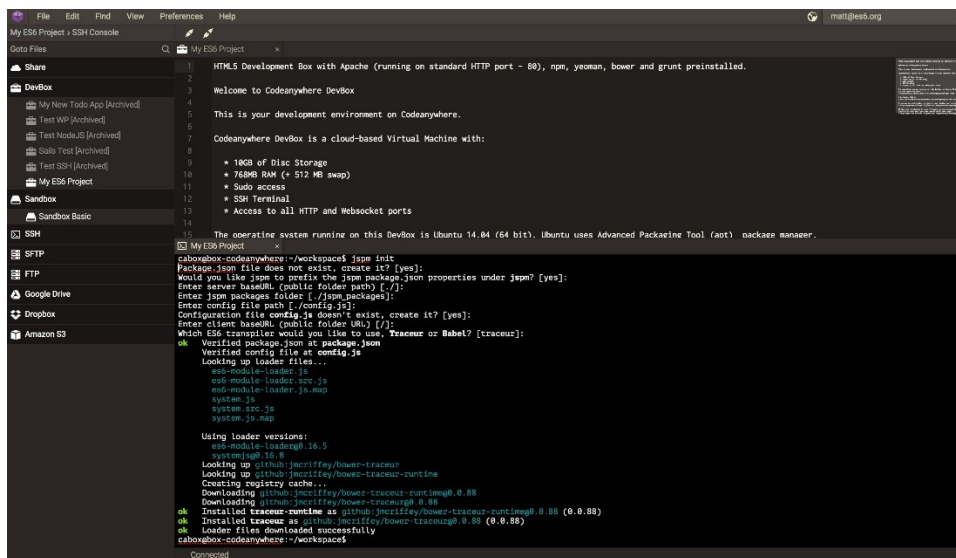
Slika 1 - PaizaCloud okruženje (PaizaCloud environment, n.d.)

5.1.2. Codeanywhere

Codeanywhere je integrirano razvojno okruženje s više različitih platformi. Omogućuje korisnicima pisanje koda kao što je vidljivo u slici 2, te pokretanje njihova projekta iz web preglednika što samim time omogućuje otvaranje i preko računala ili mobilnog uređaja.

Napisan je u cjelovitosti u programskom jeziku JavaScript te koristi uređivač i spremnike za različita razvojna okruženja. Također, korisnicima omogućuje pokretanje u vlastitim okruženjima zvanim *DevBoxes* ili se korisnici pak mogu povezati preko vlastitog *Dropbox-a* ili pak *Google Drive-a*. Codeanywhere podržava gotovo 75 programskih jezika uključujući Ruby, Python, JavaScript, PHP i druge.

U usporedbi s na primjer korištenjem okruženja koje je dostupno samo na računalima, Codeanywhere kao što mu samo ime kaže ima dostupne aplikacije za Android uređaje i samim time troši manje vremena za instalaciju. Smatra se najpopularnijim integriranim razvojnim okruženjem budući da korisnici mogu pisati, ispravljati i generalno kompajlati svoje kodove gdje god se nalaze.



```
File Edit Find View Preferences Help
My ES6 Project - SSH Console matt@es6.org
Goto Files My ES6 Project
Share
DevBox
  My New Todo App [Archived]
  Test WP [Archived]
  Test NodeJS [Archived]
  Sails Test [Archived]
  Test SSH [Archived]
  My ES6 Project
SandBox
  Sandbox Basic
SSH
  My ES6 Project
SFTP
FTP
Google Drive
Dropbox
Amazon S3

1 HTML5 Development Box with Apache (running on standard HTTP port = 80), npm, yarn, bower and grunt preinstalled.
2
3 Welcome to Codeanywhere Devbox
4
5 This is your development environment on Codeanywhere.
6
7 Codeanywhere DevBox is a cloud-based Virtual Machine with:
8
9   * 10GB of Disc Storage
10  * 768MB RAM (+ 512 MB swap)
11  * Sudo access
12  * SSH Terminal
13  * Access to all HTTP and Websocket ports
14
15 The operating system running on this DevBox is Ubuntu 14.04 (64 bit). Ubuntu uses Advanced Package Tool (apt) package manager.
16
17 My ES6 Project
18
19 cabox@codeanywhere:~/workspace$ jspm init
20 Package.json file does not exist, create it? [yes]:
21 Would you like jspm to prefix the jspm package.json properties under jspm? [yes]:
22 Enter server baseURL (public folder path) [/]:
23 Enter jspm packages folder [./jspm_packages]:
24 Enter config file path [./config.js]:
25 Configuration file config.js doesn't exist, create it? [yes]:
26 Enter client baseURL (public folder URL) [/]:
27 Which ES6 transpiler would you like to use, Traceur or Babel? [traceur]:
28 ok
29 Verified package.json at package.json
30 Verified config file at config.js
31 Looking up loader files...
32   es6-module-loader.js
33   es6-module-loader.js.map
34   system.js
35   system-amd.js
36   system.js.map
37
38 Using loader versions:
39   es6-module-loader@0.16.5
40   systemjs@0.16.0
41 Looking up @jspm/jacquire/bower-traceur:
42 Looking up @jspm/jacquire/bower-traceur-runtime
43 Creating registry cache...
44 Downloading @jspm/jacquire/bower-traceur@0.8.8
45 Downloading @jspm/jacquire/bower-traceur@0.8.8
46 ok Installed traceur-runtime as @jspm/jacquire/bower-traceur-runtime@0.8.8 (0.8.88)
47 ok Installed traceur as @jspm/jacquire/bower-traceur@0.8.8 (0.8.88)
48 ok Loader files downloaded successfully
49 cabox@codeanywhere:~/workspace$
50
51 Connected
```

Slika 2 - Codeanywhere okruženje (Codeanywhere, 2019)

5.1.3 Usporedna analiza Web 2.0 alata za učenje

U tablici 2 prikazana je usporedna analiza Web 2.0 alata za učenje programiranja prema zadanim kriterijima te je svakom dodijeljena ocjena zadovoljavanja kriterija.

WEB 2.0 ALATI ZA UČENJE PROGRAMIRANJA		
KRITERIJI	ALATI	
	PaizaCloud IDE	Codeanywhere
Zainteresiranost korisnika za učenje	DOBRO	ODLIČNO
Jednostavnost učenja	VRLO DOBRO	VRLO DOBRO
Dostupnost materijala i vodiča za učenje	DOBRO	DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	VRLO DOBRO	VRLO DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	DOBRO	ODLIČNO
Motiviranost korisnika za danjim učenjem	DOBRO	ODLIČNO

Tablica 2 - Usporedna analiza Web 2.0 alata (Rad autorice)

- **PaizaCloud IDE** – zainteresiranost korisnika za ovakvom vrstom učenja programiranja se pokazala dobrom s obzirom da je samo razvojno okruženje dosta jednostavno za učenje, a samim time i pokazuje vrlo dobro pružanje informacija o pogreškama i potencijalnim problemima s kojim se učenici ili studenti koji se odluče za ovaj alat mogu susresti. Dostupnost materijala poput stručne literature ili pak vodiča za učenje se u većini slučajeva svodi na internet izvore i razne video tutorijale. Sama motiviranost korisnika za ovakvim alatom za učenje programiranja se pokazala dobrom jer ima sve tipične značajke jednog okruženja koja se očekuju.
- **Codeanywhere** – zainteresiranost za korištenje ovog alata se pokazala odličnom samim time jer je dostupan za korištenje i preko Android uređaja što je dovelo do toga da se i motiviranost korisnika za danjim učenjem pokaže odličnom. Jednostavnost učenja pruža dodatne mogućnosti i još veću zainteresiranost korisnika, a dostupnost materijala i vodiča za učenje se kao i kod prethodnog pokazala dobrom jer se u većini slučajeva svodi na online vodiče i video tutorijale. Pruža vrlo dobre informacije o pogreškama te daje mogućnost ispravljanja istih. Pruža odlične mogućnosti i različite funkcionalnosti.

5.2. Suradničko učenje pomoću društvenih Web aplikacija

Kolaborativno odnosno suradničko učenje programiranja kao što je već prije rečeno vrlo je popularno i danas sve više zastupljeno kako među studentima tako i među vrhunskim stručnjacima i suradnicima na velikim projektima.

Kao Web 2.0. alati za suradničko programiranje najviše su se istakli blog i wiki kao alati koji danas zastupaju veliki broj internetskih sadržaja.

Najpopularnija skupina alata i razvojnih sučelja prema Škoriću (2010, 4) su:

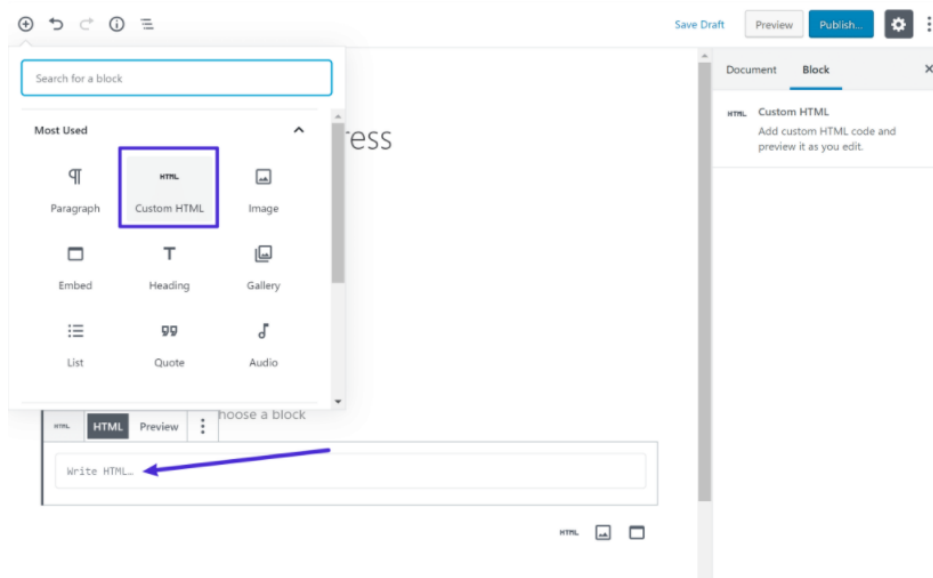
- **RIPPLE** je alat za distribuirano programiranje u paru. U takvom kontekstu studenti mogu istovremeno raditi na istom kodu i imaju mogućnosti razgovora za vrijeme pisanje koda. Istraživanja su pokazala da studenti imaju pozitivan odnos prema korištenju takvih alata.
- **CodeWave** je integrirano programsko okruženje koji omogućuje funkcionalne razvojne okoline i sve podatke koji su potrebni studentima na jednom takvom sučelju. Kroz razne poruke sa drugim korisnicima šalju se poruke i veze o određenim dijelovima koda. Za profesore ili nastavnike ovaj alat nudi mogućnost pregleda prošlosti uređivanja.
- **Jimbo** – suradničko je integrirano razvojno okruženje na webu sa velikim brojem ugrađenih alata.
- **CoRED** – omogućava suradničko kodiranje, ali nema nikakve dodatne alate.
- **AliCe-ViLlagE** – verzija poznate Alice.sa za programiranje u paru. Stvara virtualne svjetove i spaja objekte u tim svjetovima.

5.2.1. Blog

Blog (*eng. Weblog*) je oblik objavljivanja informacija na internetu u kojem se prikazuju prvenstveno periodični članci u obrnutom vremenskom slijedu što bi značilo da se najnoviji članci samim time nalaze na vrhu stranice. Blog kao takav služi za stvaranje i objavljivanje sadržaja u velikom broju slučajeva već programiranih sučelja koji se vrše preko standardnih web preglednika i od samih korisnika ne zahtjeva nikakvo posebno znanje programiranja što bi značilo da se takozvanim "*bloganjem*" mogu baviti i ljudi koji ne znaju ništa o programiranju.

Ipak, umjesto kupnje već izrađenog sučelja iza kojeg stoje programi kao što su HTML5 ili PHP, na većini platformi za izradu bloga omogućeno je uređivanje ili dodavanje vlastitih kao što je vidljivo na slici 3, te time značajno mijenja mišljenja o blogu kakav je naveden prije. Samim time, upravo je to jedan od savršenih načina za ulazak u svijet programiranja i samim time blog kao takav je jedan od najboljih alata za suradničko programiranja jer se u većini slučajeva korisnici odlučuju za rad u paru.

Upravo na taj način su Bubaš et al. (2012) u svom istraživačkom radu i komunikaciji sa studentima proveli korištenje *Wordpress-a* kao alata za vođenje bilješki za vrijeme predavanja na fakultetu u koji su trebali pohraniti tekst, fotografije i slično.



Slika 3 - Uređivanje HTML-a u okruženju Wordpress (Kinsta, 2020)

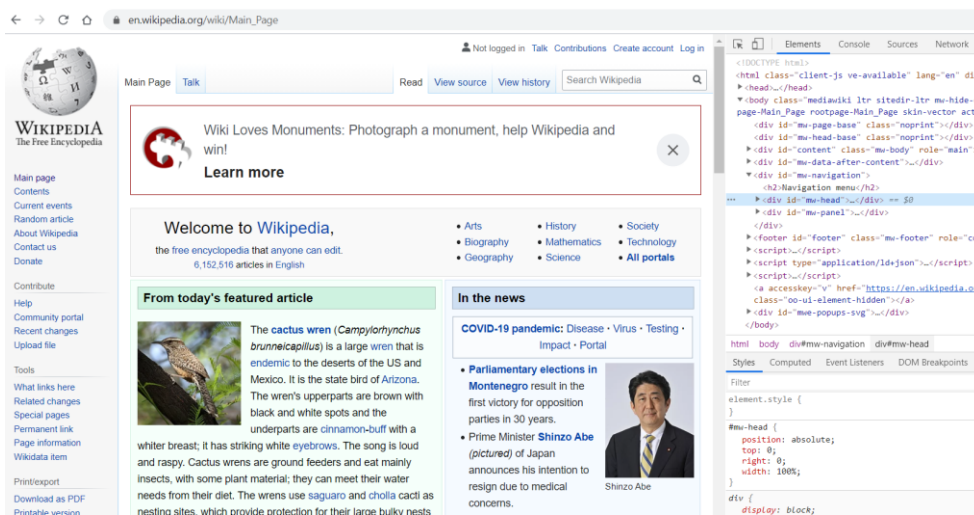
5.2.2. Wiki

Wiki kao alat stvoren je za zajedničko pisanje i suradnju te je savršen primjer na kojem se temelji i zasniva Web 2.0. Najveći dio sadržan je u samoj ideji da se što većem broju korisnika omogući da kreiraju isti zajednički sadržaj te uređuju, prepravljaju i nadovezuju se na ono već napisano ili u slučaju pogrešnih ili krivo interpretiranih i napisanih informacija ih brišu.

Wiki je po svojim karakteristikama vrlo pogodan za različite projekte ili dijeljenje razmišljanja, izmjenu i razvoj ideja među članovima tima ili u paru te su s godinama veliku praksu pronašli u poslovnim krugovima diljem svijeta te se danas u većini slučajeva kao potpomognut alat koristi i u učenju. Nadalje, iako je wiki pun informacija, na žalost njegov temelj, a to je otvorenost i mogućnost otvorenog uređivanja mu je i mana jer ipak se radi o web mjestu na kojem svi mogu uređivati tekst o kojem se piše, bili oni laici ili vrsni stručnjaci o nekoj temi.

Od velike je važnosti provjeriti svaku točnost, ali kao alat koji eventualno služi kako bi se nešto kreiralo i vodilo po tome jako dobro funkcionira. Wiki je jedna od velikih baza podataka koja koristi PHP što se vidi na slici 4.

Koristi *MediaWiki softver*, a kao bazu podataka *MySQL*.



Slika 4 - Wikipedia okruženje (Izgled početne stranice weba (wikipedia.org) prikazana na računalu)

5.2.3. Usporedna analiza alata za suradničko učenje na Web-u

U tablici 3 prikazana je usporedna analiza alata za suradničko učenje programiranja prema zadanim kriterijima te je svakom dodijeljena ocjena zadovoljavanja kriterija.

WEB 2.0 SURADNIČKO UČENJE PROGRAMIRANJA		
KRITERIJI	ALATI	
	Blog	Wiki
Zainteresiranost korisnika za učenje	ODLIČNO	DOBRO
Jednostavnost učenja	VRLO DOBRO	ODLIČNO
Dostupnost materijala i vodiča za učenje	DOBRO	DOVOLJNO
Pružanje informacija o pogreškama i potencijalnim problemima	DOVOLJNO	DOVOLJNO
Pružanje različitih mogućnosti i funkcionalnosti	VRLO DOBRO	DOVOLJNO
Motiviranost korisnika za danjim učenjem	VRLO DOBRO	DOBRO

Tablica 3 - Usporedna analiza Web 2.0 alata za suradničko učenja (Rad autorice)

- **Blog** – zainteresiranost korisnika za učenjem programiranja preko bloga se pokazala odličnom pogotovo jer uređivanjem HTML-a mogu stvarati vlastite teme. Jednostavnost učenja, pružanje različitih mogućnosti i funkcionalnosti te motiviranost korisnika za daljnjim učenjem se pokazala vrlo dobrom prvenstveno zbog različitih načina poslovanja i kreiranja web stranica. Blog kao takav i van uređivanja HTML-a pruža mnoštvo mogućnosti za uređivanje teksta, prijenos fotografija i slično. Dostupnost materijala za izradu bloga i uređivanje danas su dostupne većoj publici i to uglavnom u online obliku preko različitih foruma ili pak video uradaka. Ipak, pružanje informacija o pogreškama i potencijalnim problemima je samo dovoljno i potreban je velik broj dodataka.
- **Wiki** – zainteresiranost korisnika za učenjem programiranja preko alata wiki u smisli uređivanja različitih informacija se pokazala dobrom, ali samog cilja programiranja slabom iako je motiviranost korisnika također dobra. Jednostavnost učenja kao suradničkog alata se pokazala odličnom, no dostupnost materijala i vodiča za učenje kao i pružanje različitih mogućnosti i funkcionalnosti na samoj platformi je dovoljna za osnovne potrebe i početak učenja programiranja. Pružanje informacija o pogreškama i potencijalnim problemima nakon objave ipak daju drugi korisnici, a ne sam alat.

6. Vizualni alati za učenje programiranja

Razvitak tehnologije danas donosi veliki napredak tehnologije u svim granama znanosti u smislu medicine, bio znanosti, fizike i astronomije. Napredak koji tehnologija daje se vidi na svakom koraku i svakidašnji život time raste, a samim time raste i potreba za usavršavanjem. Danas praksa i praktični radovi puno više znače i imaju korist od samog čitanja knjiga, tako da bilo to stručno osposobljavanje, učenje programiranja na fakultetima ili možda samoučenjem, dolazi se do pravih stručnjaka koji tehnologiju danas vode u kompletno drugim smjerovima no što se zamišljalo.

Uz već spomenutu motivaciju i organizaciju, ljudima je danas za ključno učenje bitna zanimacija, odnosno da im je nešto zanimljivo i da se "vide u tome". Svako gradivo koje studenti ili učenici prolaze na fakultetima ili možda kroz samoučenje mora biti prilagođeno njihovom stupnju predznanja ili znanja koje su konkretno stekli do trenutka kada se odlučuju za neki projekt. Mnogim istraživanjima je zaključeno da skoro 58% ljudi na svijetu pamti vizualno, a vizualni alati za učenje programiranja i pamćenje ključni su kako bi se osnovnoškolce i srednjoškolce zajedno sa studentima pripremilo na teže i zahtjevnije učenje programiranja.

Najznačajniji u svemu je pristup i kontekst učenja koji spaja nastavnike i profesore sa sadržajem, ali isto tako i učenike s nastavnicima, a same učenike sa sadržajem koji uče i koji ih zanima.

Jedni od **najpopularnijih vizualnih alata** za učenje programiranja su svakako:

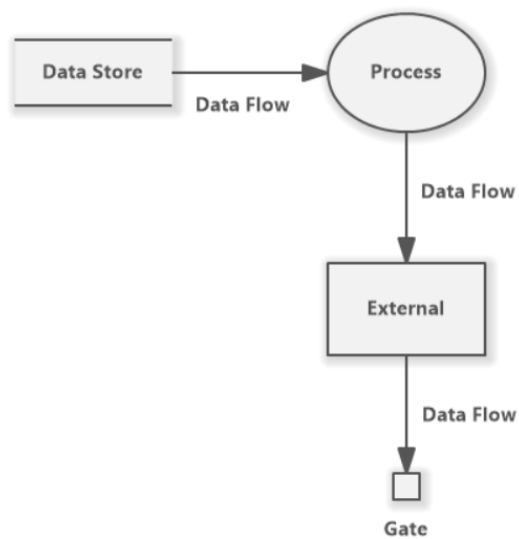
- alati namijenjeni učenju kroz dijagram toka
- posebni alati za vizualizaciju algoritma
- mini-jezici namijenjeni konkretnom učenju mlađih uzrasta kako bi se stekla programska logika i logičko-matematički način razmišljanja

6.1. Alati za učenje kroz dijagram toka

Dijagram toka ili tijeka (*eng. flow chart*) simbolički je algoritam koji se sastoji od simbola koji su povezani strelicama, odnosno nekim vanjskim "putevima" koji na kraju definiraju sam tijek i smjer programa kao što je prikazano na slici 5.

Osnovni elementi sastoje se od:

- Početka programa
- Ulaza
- Čvorišta
- Obrade
- Petlje
- Petlje i blokova
- Izlaza



Slika 5 - Primjer dijagrama toka (*Software ideas modeler, 2020*)

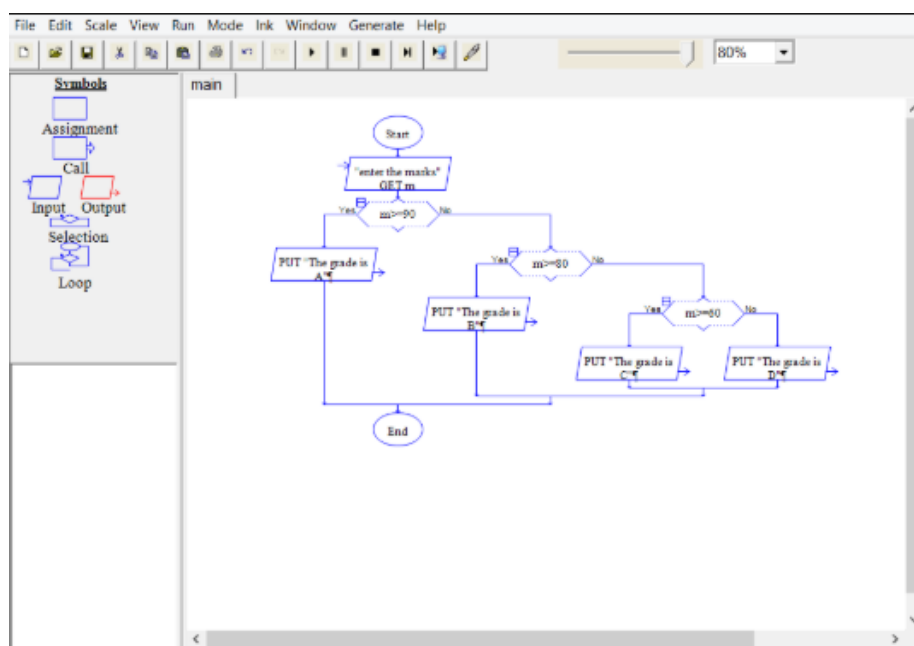
Alati poput dijagrama toka mogu ubrzati rješenje problema ili nekog procesa s tim da u isto vrijeme mogu povećati razumijevanje problema te vizualno granati probleme na manje probleme. Generalno, prikazuju tijek događanja i daju kompletnu sliku cijelog programa, ali napravljen na manje sektore odnosno dijelove.

6.1.1. RAPTOR

RAPTOR je posebno dizajnirano programsko okruženje koje se temelji na izradi dijagrama toka, odnosno stvoreno kako bi se korisnicima vizualno prikazao algoritam kao što prikazuje slika 6 i samim time se smanjile eventualno nastale greške u pisanju koda.

Samim time, RAPTOR smanjuje ukupnu sintaksu koju korisnici moraju naučiti kako bi na kraju njihov program mogao raditi samostalno i najbitnije ispravno radio. Kao i kod svakog dijagrama toka tako i u okruženju RAPTOR simboli koji su prikazani se izvršavaju jedan za drugim pa je samim time i logičnije praćenje kako program radi.

RAPTOR kao takav ima mogućnost i prikazuje greške što korisnicima uvelike olakšava praćenje izvršavanje koda te ga čini jasnijim i jednostavnijim za veće projekte. Čest je u korištenju za učenje jer ima jednostavan jezik i grafičke komponente te na taj način na dobar način podučavaju korisnike tj. učenike ili studente u glavnim idejama računalnog programiranja.

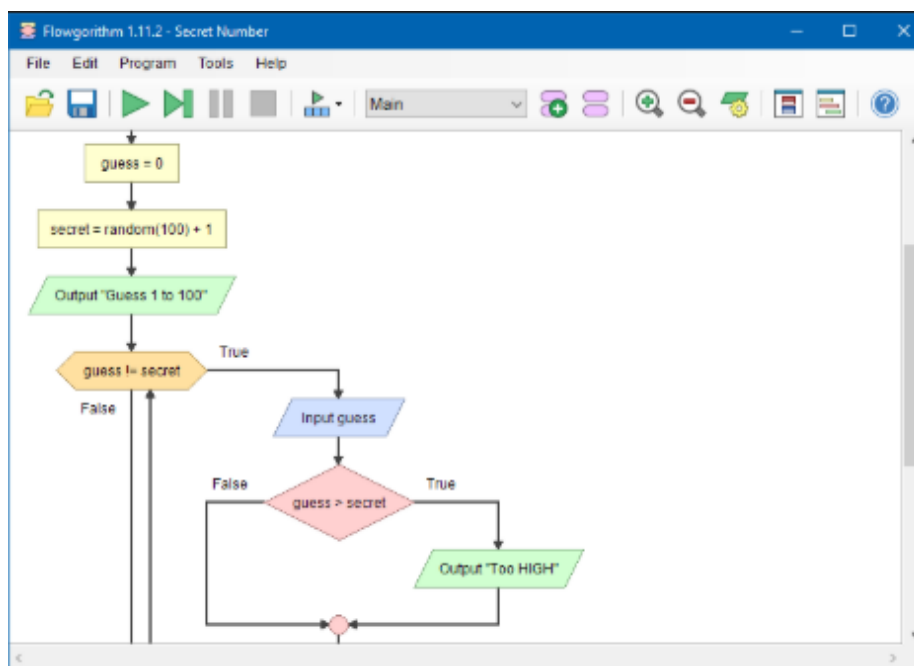


Slika 6 - Izrada dijagrama toka u okruženju RAPTOR (GeeksforGeeks, 2018)

6.1.2. FlowGorithm

FlowGorithm je grafički alat koji služi korisnicima za autorizaciju, pisanje i izvršavanje programa, ali pomoću dijagrama toka kao što je prikazano na slici 7. Osmišljen je tako da bi mu se zapravo naglasio sami algoritam, a ne sintaksa bilo kojeg programskog jezika za koji se korisnik odluči (C++, Python, Java i sl.). Kombinacija je klasičnih simbola karakterističnih za dijagram toka, ali i onih simbola koje koriste druge vrste dijagrama. Također, različite boje se dijele za svaki generirani kod, ali i sami prozor konzole. Podržava rekurzije, petlje, nizove i različite izraze.

Samim time razni protokoli podataka sadrže vrlo razumljive izlaze i grafički promjenjive varijable te generiranje koda na 16 svjetskih jezika. Ima mogućnost mijenjanja već ugrađenih tema.



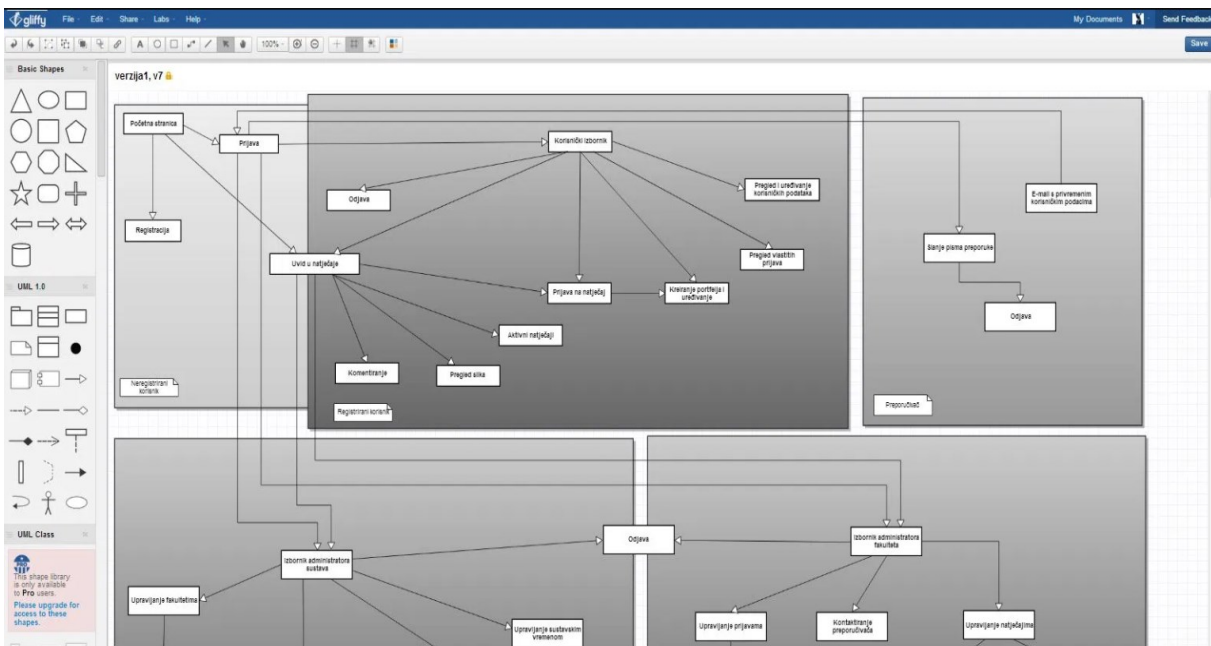
Slika 7 - Izrada dijagrama toka u okruženju FlowGorithm (Software informaer, n.d.)

6.1.3. Gliffy

Gliffy je alat za izradu dijagrama toka preko preglednika. Temeljen je na HTML5 te podržava izradu različitih dijagrama tokova, organizacijskih dijagrama koji su pokazani na slici 7, pa sve do tehničkih crteža i skica aplikacijskog sučelja.

Podržava različite teme i predloške, ima mogućnost uključivanja slika i datoteka na same dijagrame. Namijenjen je višim razredima i srednjoj školi, a odličnim se pokazuje i za fakultete, no ne samo one koji su usmjereni na informatiku i računarstvo nego i ekonomiju i strojarstvo.

Mogućnost da više korisnika radi na istom dijagramu nije moguća, plaća se za više mogućnosti rada, ali i u besplatnoj verziji mogu se dobiti kvalitetni dijagrami.



Slika 8 - Izrada dijagrama toka u okruženju Gliffy (E-laboratorij, 2014)

6.1.4. Usporedna analiza alata za izradu dijagrama toka

U tablici 4 prikazana je usporedna analiza alata za učenje kroz dijagram toka prema zadanim kriterijima te je svakom dodijeljena ocjena zadovoljavanja kriterija.

ALATI ZA UČENJE PROGRAMIRANJA KROZ DIJAGRAM TOKA			
KRITERIJI	ALATI		
	RAPTOR	FlowGorithm	Gliffy
Zainteresiranost korisnika za učenje	ODLIČNO	VRLO DOBRO	ODLIČNO
Jednostavnost učenja	VRLO DOBRO	DOBRO	DOBRO
Dostupnost materijala i vodiča za učenje	VRLO DOBRO	VRLO DOBRO	ODLIČNO
Pružanje informacija o pogreškama i potencijalnim problemima	DOBRO	DOBRO	VRLO DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	VRLO DOBRO	VRLO DOBRO	VRLO DOBRO
Motiviranost korisnika za danjim učenjem	ODLIČNO	DOBRO	ODLIČNO

Tablica 4 - Usporedna analiza alata za učenje kroz dijagram toka (Rad autorice)

- **RAPTOR** – zainteresiranost za izradu dijagrama toka pokazala se odličnom. Jednostavnost učenje i dostupnost materijala i vodiča se osim na internetu putem foruma i video tutorijala može naći i u stručnim literaturama. Također, pružanje različitih mogućnosti i funkcionalnosti je vrlo dobro jer daje logičan slijed izrade i lak je za pratiti izvršenje programa. Pružanje informacija o pogreškama i potencijalnim problemima se pokazala dobrom u vidu praćenja za vrijeme izvršnog koda. Motiviranost studenata za daljnjim učenjem je vrlo dobra zbog same pristupačnosti i dostupnosti programa.
- **FlowGorithm** – sličan prethodnom razvio je vrlo dobru zainteresiranost među studentima za učenjem. Također i dostupnost materijala i vodiča koji su dostupni kako na internetu tako i u stručnim literaturama. Pruža i vrlo dobre različite mogućnosti i funkcionalnosti. Uz ostale alate za izradu dijagrama toka pokazao se dobar i jednostavan za učenje te pružanje informacija o pogreškama i potencijalnim problemima. Uz sve ostale alate koji su dostupni pokazao se dobrim za motiviranost korisnika za daljnjim učenjem.

- **Gliffy** – s obzirom da se radi o online softveru za izradu dijagrama tokova i raznih drugih dijagrama, pokazao se vrlo motivirajućim za srednjoškolce i studente u područjima informatike, računarstva i ostalih društvenih znanosti. Iako se sve njegove opcije naplaćuju ima odličnu mogućnost pokazivanja potencijalnih grešaka i problema i bez plaćanja licence. Zbog velike količine internet izvora te jednostavnosti učenja iznimno je pogodan za korištenje u školama za edukaciju, naravno ako se ustanove i škole odluče za korištenje upravo njega u edukacijske svrhe. Motivacija za daljnjim učenjem se pokazala odličnom.

6.2. Alati namijenjeni vizualizaciji algoritma

Još jedna vrsta alata koja ulazi u skupinu vizualnih alata za učenje programiranja. Omogućuje svladavanje kompletnih problema te pomaže studentima u programiranju. Upravo takvi alati pomažu u različitim vrstama sortiranja i time daju vizualni uvid u to kako se sortiranje odvija.

Vrste sortiranja:

- Mjehurićasto sortiranje
- Koktel sortiranje
- Sortiranje odabirom
- Sortiranje umetanjem
- Višestruko sortiranje umetanjem
- Sortiranje češljanjem
- Sortiranje pomoću hrpe

Vizualizacijskim alatima se korisnicima omogućuje stvaranje programa grafičkim elementima umjesto tekstualnih. I upravo njima se učenicima ili studentima omogućuje lakša vizualizacija različitih vrsta algoritama.

Najistaknutiji alati su:

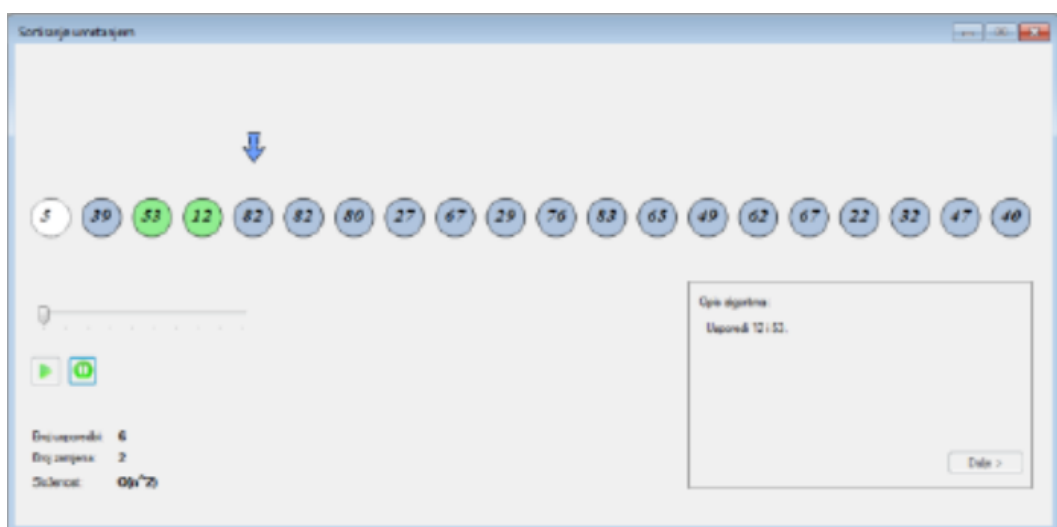
- ViSA
- PITON

6.2.1. ViSA

U svom radu Reif i Orehovački (2012) opisuju alat nazvan skraćenicom ViSA koji pomaže studentima u vizualizaciji algoritma za sortiranje. Usporedbom algoritma sortiranja te jednostavnim ponavljanjem i automatskim sustavom za vizualizaciju ViSA je jedan od najboljih algoritama namijenjenih vizualizaciji to jest, alat za učenje programiranja, vrlo jednostavan za korištenje sa jednostavnim sučeljem te je jednostavan za učenje jer pruža puno povratnih informacija i jednostavan je za učenje. Velika brzina integracije te brzi pristup koji samo jednim klikom dolazi do spremnosti za upotrebu.

Nakon što se odabere jedan od algoritama korisniku će se ponuditi određeni obrazac u kojem se definira tekstualni okvir, pregled polja i oznake kao što je prikazano na slici 9. Korisnik može izabrati također i jednu metodu generiranja na primjer brojeva. Ako na primjer korisnik unese neku vrijednost, to će učini putem praznih polja, što će za rezultat imati pokretanje polja.

Kada korisnik završi s implementacijom svih podataka, kreće pokretanje konačnog testa unesenih podataka te na taj način omogućuje korisniku nastavak programiranja. Na taj se način omogućuje korisniku da provjerava svoj rad pojedinačno po segmentima, kako bi na kraju bio siguran da dijelovi samog algoritama ispravno funkcioniraju prilikom izvedbe programa.



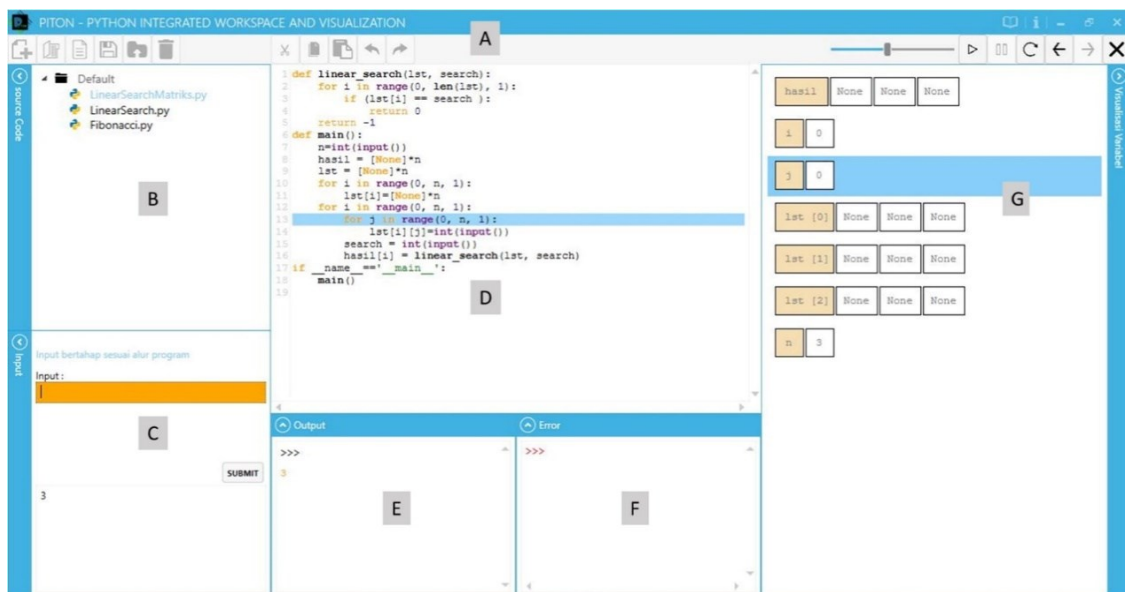
Slika 9 - ViSA sortiranje (Reif I. i Orehovački T., 2012, str. 3)

6.2.2. PITON

PITON (*Python Integrated Wokspace and Visualization*) je prema Elvini et. al. (2018) edukacijski programski alat koji kombinira radni prostor za programiranje s alatom za vizualizaciju.

Stoga PITON ne omogućuje samo izravni razvoj izvornog koda nego i vizualizaciju samog koda kao što je vidljivo na slici 10. Cilj mu je prvenstveno pomoći korisnicima (najčešće studentima) u programskom jeziku Python.

Sadrži veliki broj značajki i za razliku od ViSA-e koristi se za samu vizualizaciju programa neovisno radi li se samo o sortiranju nekog od algoritama ili o kompletnom programu. Upravo zbog toga uvelike pomaže studentima u njihovim projektnim zadacima.



Slika 10 - Vizualizacija u alatu PITON (E. Elvina et. al., 2018, str. 268)

6.2.3. Usporedna analiza alata namijenjenih vizualizaciji algoritma

U tablici 5 prikazana je usporedna analiza alata namijenjenih virtualizaciji algoritma prema zadanim kriterijima te je svakom dodijeljena ocjena zadovoljavanja kriterija.

ALATI NAMIJENJENI VIZUALIZACIJI ALGORITMA		
KRITERIJI	ALATI	
	ViSA	PITON
Zainteresiranost korisnika za učenje	ODLIČNO	ODLIČNO
Jednostavnost učenja	VRLO DOBRO	VRLO DOBRO
Dostupnost materijala i vodiča za učenje	DOBRO	DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	DOBRO	DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	DOBRO	VRLO DOBRO
Motiviranost korisnika za danjim učenjem	VRLO DOBRO	VRLO DOBRO

Tablica 5 - Usporedna analiza vizualizacijskih alata (Rad autorice)

- **ViSA** – zainteresiranost korisnika, izričito studenata za ovim alatom je odlična. Jednostavnost učenja se pokazala vrlo dobrom ako su korisnici upoznati sa različitim vrstama sortiranja. Dostupnost materijala i vodiča za učenje se svodi na internet izvore i različite recenzije i znanstvene članke, no bez obzira na to studentima i korisnicima općenito je motiviranost za danjim učenjem u tom slučaju vrlo dobra. Ipak, pružanje informacija o pogreškama i potencijalnim problemima kao i pružanje različitih mogućnosti i funkcionalnosti se pokazala dobrom što je sasvim solidno s obzirom da je namijenjena upravo vizualizaciji algoritma sortiranja.
- **PITON** – zainteresiranost studenata koji programiranje provode u Python-u je odlično jer ovaj alat upravo služi za vizualizaciji programa upravo u njemu. Jednostavnost učenja je vrlo dobra s obzirom da se pretpostavlja da studenti već znaju raditi u programskom jeziku, no i ako ne znaju postoji dobra dostupnost materijala i vodiča za učenje (posebno na internetu). Alat nudi vrlo velik broj različitih mogućnosti i funkcionalnosti kao i pružanja informacija o potencijalnim problemima koji se mogu stvoriti što je vrlo dobro. Motiviranost najčešće studenata za danjim učenjem je vrlo dobra jer PITON prikazuje vizualizaciju cijelog napisanog programskog koda što uvelike pomaže.

6.3. Mini-jezik kao alat za učenje programiranja

Mini-jezici su grupa vizualnih jezika koji sadrže određene alate kojima se može pomoći u početnom učenju programiranja.

Osnovna ideja takvih jezika je pristupačnost u svijetu programiranja i stvaranje dobrog i pogodnog dizajna koji će početnicima olakšati svladavanje problema kroz nizove naredbi koje će dati računalu, a zauzvrat dobiti neki grafički prikaz. Također, osnova je za stvaranje temeljnog znanja i temeljnih vještina te razvijanje programske logike kako bi se kasnije moglo prelaziti na druge programske jezike.

6.3.1. MSWLogo

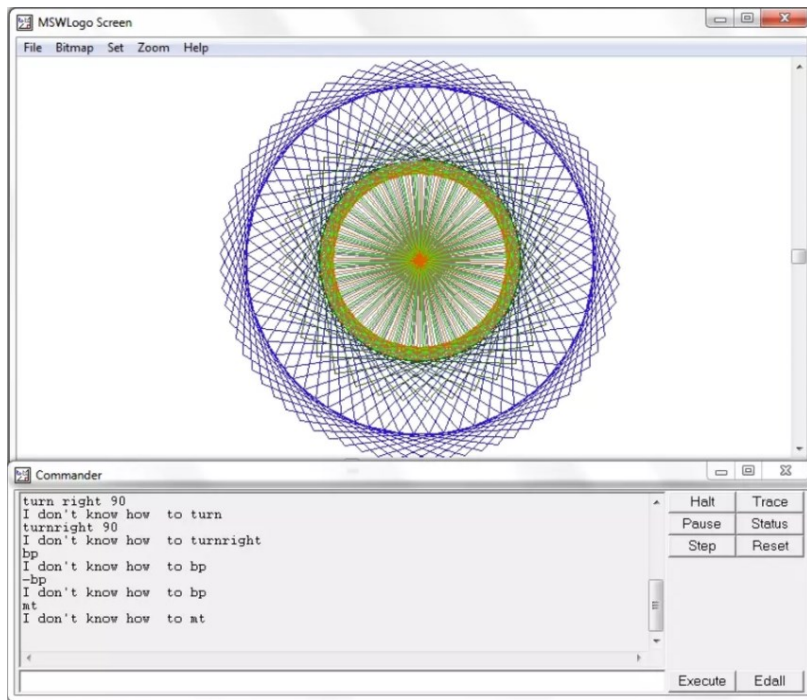
MSW Logo je programski jezik, odnosno mini-jezik koji je prvobitno bio konstruiran kako bi se djecu uvelo u svijet programiranja.

Cilj mu je prije svega bio stvaranje i razvijanje programske logike. Logo kao takav je bio namijenjen djeci i njegova najveća namjena je računalna grafika. Istraživanjem je Logo postao jezik širokog opsega i mogli su ga koristiti učenici u različitim dobnim skupinama, odnosno različitim razredima na primjer osnovne škole.

U prošlosti je definitivno najpopularnija verzija bila ona koja je u sebi imala geometrije kornjače. Djeca su zapravo samim time koristili kornjaču kako bi istraživali različite geometrijske oblike.

Dakle, Logo je djeci svakako omogućio pisanje jako jednostavnih kodova za stvaranje različitih oblika kao što je prikazano na slici 11. Za razliku od drugih programskih jezika Logo u potpunosti ima takozvanu uniformnu sintaksu što bi značilo da on razumije različite naredbe, a one mogu biti postavljene na puno različitih načina.

Kao i ostali programski jezici zahtjeva vrijeme kako bi se shvatio, no sam ulazak u svijet i program s kojim većina djece kreće u informatički svijet pruža jako dobar temelj u učenju programiranja te samim time daje motivaciju i zanimaciju kako bi djeca nastavila u tom smjeru.



Slika 11 - MSWLogo okruženje (Uptodownd, n.d.)

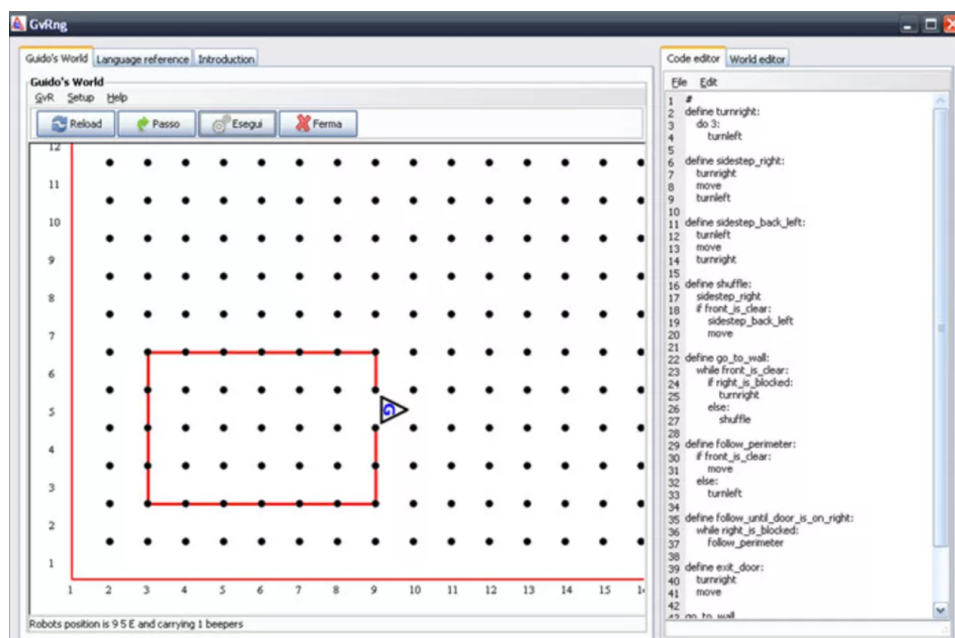
6.3.2. Guido van Robot

Guido van Robot bazira se na minimalizmu programskog jezika te pruža dovoljno sintakse različitim korisnicima, učenicima ili studentima kako bi naučili pojmove poput grananja, petlji ili proceduralne apstrakcije.

Najveća i najznačajnija stavka ovog alata je učenje programiranja u okruženjima koje rješavaju probleme uz povratne informacije i to one vizualne. Dakle, njegova najveća prednost je u tome što dopušta učenje programiranja u okruženju koje kombinira rješavanje problema sa rezultatima u obliku vizualnih povratnih informacija

Generalno je interaktivan program koji je izvrstan za učenje osnovnih pojmova programiranja poput sekvenciranja, procedura i uvjetnih grananja. Kao takav, dostupan je za Mac i Windows platforme, a omogućuje surađivanje sa programskim jezicima poput Python-a koji se također pokazao kao vrlo dobar program za učenje programiranja.

Kreirali su ga upravo studenti i suština mu je robot koji je predstavljen sa trokutom, što se može vidjeti na slici 12. koji je okružen različitim barijerama te su njegove akcije vođene isključivo onime što je korisnik napisao.



Slika 12 - Guido van Robot okruženje (Softonic, n.d.)

6.3.3. Usporedna analiza mini-jezika

U tablici 6 prikazana je usporedna analiza mini-jezika prema zadanim kriterijima te je svakom dodijeljena ocjena zadovoljavanja kriterija.

MINI-JEZIK KAO ALAT ZA UČENJE PROGRAMIRANJA		
KRITERIJI	ALATI	
	MSWLogo	Guido van Robot
Zainteresiranost korisnika za učenje	ODLIČNO	ODLIČNO
Jednostavnost učenja	ODLIČNO	VRLO DOBRO
Dostupnost materijala i vodiča za učenje	ODLIČNO	VRLO DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	DOBRO	VRLO DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	DOBRO	VRLO DOBRO
Motiviranost korisnika za daljnjim učenjem	VRLO DOBRO	VRLO DOBRO

Tablica 6 - Usporedna analiza mini-jezika (Rad autorice)

- **MSWLogo** – zainteresiranost mlađih skupina za ovim mini jezikom je odlična, posebice kad se prvi put susreću s programiranjem. Također, jednostavnost učenja i dostupnost materijala i vodiča za učenjem je odlična osobito jer se i školskom kurikulumu za nastavu informatike (https://narodne-novine.nn.hr/clanci/sluzbeni/2018_03_22_436.html, 2020) nalazi MSWLogo. Pružanje informacija o pogreškama i potencijalnim problemima i pružanje različitih mogućnosti i funkcionalnosti su dobre u edukacijske svrhe i osobito za korisnike (učenike u osnovnim školama) koji se tek upoznaju s programiranjem. Samim time je motiviranost korisnika za daljnjim učenjem vrlo dobra i stvara dobru potporu i razvoj logičkog razmišljanja kako bi se kasnije prešlo na kompleksnije i zahtjevnije programe.
- **Guido van Robot** – zainteresiranost korisnika za ovim mini programskim jezikom je odlična. Također jednostavnost učenja je vrlo dobra uključujući u obzir da su u vrlo dobrom smislu dostupni materijali i vodiči za učenje. Pružanje informacija o pogreškama i potencijalnim problemima je jasno vidljiv što je vrlo dobro, dok su pružanje različitih mogućnosti i funkcionalnosti odlične osobito je Guido van Robot pruža mogućnost i kompleksnijih kodova koji sadrže na primjer petlje. Kod motiviranosti korisnika za daljnjim učenjem se pokazao kao vrlo dobar.

7. Igre za učenje programiranja

Učestalost pitanja o tome može li se naučiti programirati kroz igrice dostupne na internetu su zadnjih nekoliko godina sve učestalija i ozbiljnija. Nažalost, većina ljudi misli da je programiranje kao takvo jedan veliki proces i postupak koji je kompliciran i zamršen te da oduzima jako puno vremena i odricanja kako bi se napisao nekakav program, stvorila aplikacija ili napravila nekakva online igrica. Upravo takvim gledanjem na programiranje i imajući takav način razmišljanja, u većini slučajeva dovodi mlade ljude do odustajanja prije nego krenu s programiranjem i u njima stvara strah od istog. Odustajanje od programiranja u tome trenutku je izuzetno velika, a cilj alata namijenjenih učenju programiranja je upravo to da na zanimljiv i jednostavan način djeci i ostalim korisnicima prikažu da se kroz različite naredbe koje se mogu dati računalu zauzvrat može dobiti ono što su naumili.

Prvenstveno, cilj je kroz igru te zanimljiv i interaktivan način djeci približiti programiranje i uputiti ih da kroz zabavu mogu stvoriti vrlo veliko znanje. Samim time je i vjerojatnost da će nakon nekog vremena takve vrste učenja, nastaviti se baviti programiranjem u složenijim i zahtjevnijim programima.

Orehovački et. al. (2015, 620-631) svoje istraživanje usmjerili su na studente kroz pitanje hoće li im završetak neke igre stvoriti sreću ili neki drugi osjećaj. Istraživanje je dakle napravljeno na studentima koji su u jednu ruku već odrasle osobe i koji su u odrastanju i sazrijevanju već otprilike trebali shvatiti da poanta nije uvijek u pobjedi nego bi im glavni cilj trebao biti istraživanje i učenje. Prvobitni cilj cijelog istraživanja je bio da studenti pobjede strah od programiranja te sami shvate kako upornost i rad dovode do izgradnje samopouzdanja i znanja, te kako programiranje može biti zabava i jednostavnost.

Takav način učenja programiranja kroz igre kako kod mlađih generacija tako i kod onih starijih (studentata) pokazuje zadovoljstvo nakon ispravno napisanog koda i odigrane igre. Samim time uvelike pomaže u motivaciji i želji za nastavkom učenja programiranja što i je cilj kod tako osmišljenih igri.

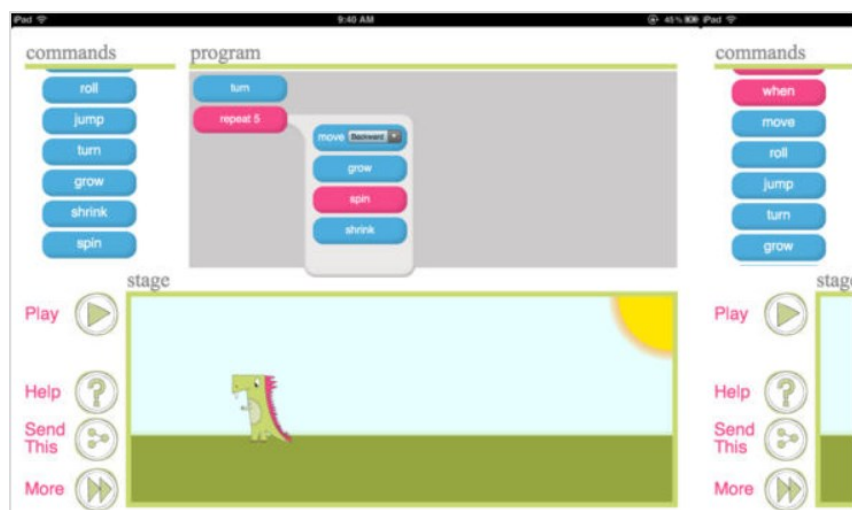
7.1. Igre za djecu

Igre namijenjene mlađim skupinama potiču prije svega jednostavnost programiranja i maštu, ali u velikoj mjeri i znatizelju djece kako bi jednom nakon svladavanja straha i jednostavnih zadataka krenuli sa zahtjevnijim i složenijim igrama. Igre za djecu su također podijeljene u skupine koje su namijenjene baš mlađim generacijama. Potiču dobru količinu programske logike i prije svega traže razumijevanje onih komandi koje se daju računalu u poprilično kratkom vremenskom periodu kako bi se svladala osnovna znanja. Upravo je to najjednostavniji način za ulazak u svijet programiranja.

7.1.1. *Daisy the dinosaur*

Daisy the dinosaur edukacijska je igra usmjerena podučavanju djece u učenju jednostavnog načina programiranja. Besplatna je i zabavna aplikacija koja se sastoji od *drag&drop* sučelja te samim time uvodi djecu u osnove programiranja jer aplikacija podržava različita sekvenciranja i grananja koja su ključna u rješavanju problema u programiranju.

Umjesto tipkanja izvornog koda koje je za djecu tog uzrasta ipak preteško, bavi se vizualnim povlačenjem blokova koda što je vidljivo na slici 13, koji se na kraju spajaju zajedno. Sami vizualni dizajn i načela se temelje na slobodnom Scratch-u. Baš kao i ostale igrice temelji se na HTML-u 5 i JavaScript-u te se može koristiti na Apple mobilnim i tablet uređajima.



Slika 13 - Daisy the dinosaur okruženje (Educational App Store, n.d.)

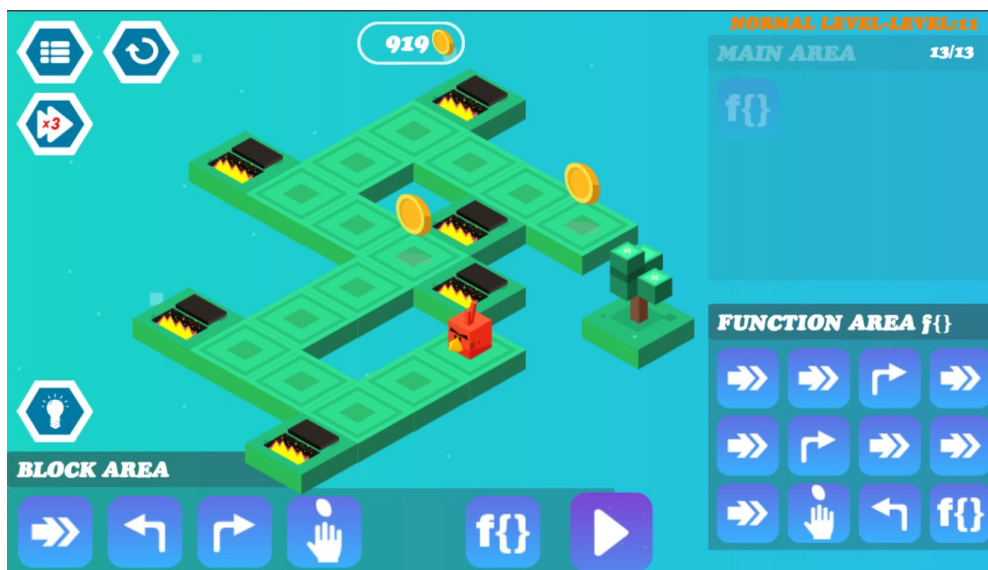
7.1.2. Algorithm city: Coding game

Algorithm city zabavna je i inovativna igra za kodiranje. Djecu mlađih dobnih skupina podučava osnovama programiranja i algoritama. Stjecanjem osnovnih pojmova poput sekvenciranja naredbi, različitih funkcija i petlji što je vidljivo na slici 14. Motivira djecu za prelazak u nove razine koje postaju teže i kompliciranije.

Samim time, odličan je uvod u svijet programiranja.

Sadrži 51 razinu koje su podijeljene u 4 poglavlja:

- *Educational Chapter* – sadrži 6 razina i bazno uči kako raditi
- *Easy Chapter* – sadrži 15 razina i podučava osnovama kodiranja
- *Normal Chapter* – također sadrži 15 razina i podučava petljama pomoću funkcija
- *Hard Chapter* – također, 15 razina i podučava složenijim funkcijama



Slika 14 - Algorithm City: Coding game okruženje (Google Play, Algorithm City: Coding Game, n.d.)

7.1.3. Usporedna analiza igri za djecu

IGRE ZA UČENJE PROGRAMIRANJA - mlađe skupine		
KRITERIJI	ALATI	
	Daisy the dinosaur	AC: Coding game
Zainteresiranost korisnika za učenje	DOBRO	ODLIČNO
Jednostavnost učenja	ODLIČNO	VRLO DOBRO
Dostupnost materijala i vodiča za učenje	DOBRO	DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	DOVOLJNO	DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	DOBRO	VRLO DOBRO
Motiviranost korisnika za danjim učenjem	DOBRO	VRLO DOBRO

Tablica 7 - Usporedna analiza igri za djecu (Rad autorice)

U tablici 7 prikazana je usporedna analiza igri namijenjenih djeci tj. mlađim skupinama prema zadanim kriterijima iz koje slijedi:

- **Daisy the dinosaur** – zainteresiranost za učenjem pokazala se dobrom, zbog jednostavnosti učenja koja je odlična. Ipak, pružanje informacija o pogreškama i potencijalnim problemima pokazala se samo dovoljnom i samim time je motiviranost korisnika za danjim učenjem ostala samo na dobrom. Dostupnost materijala i vodiča za učenje je dobra i može se uglavnom naći na recenzijama dostupnima na internetu. Mana ove igre je definitivno to što je dostupna samo u Apple trgovini.
- **Algorithm City: Coding game** – zainteresiranost korisnika (najčešće mlađih skupina djece) za ovakvom vrstom igrice se pokazala odličnom. Jednostavnost učenja je vrlo dobra kao i pružanje različitih mogućnosti i funkcija (uključujući petlje i funkcije). Dostupnost materijala i pružanje informacija o pogreškama i potencijalnim problemima se pokazala dobrom za djecu. Motiviranost djece za korištenje i daljnje učenje programiranja je vrlo dobro.

7.2. Igre za studente

Kako su stvorene igre namijenjene mlađim skupinama, tako postoje i igre namijenjene učenju programiranja malo starijih generacija, najčešće studenata. Kompliciranije su, zahtijevaju više pažnje i više programske logike od onih namijenjenih djeci, ali i dalje su poprilično jednostavne i samim time odličan uvod u svijet programiranja u kompleksnijim programima koji zahtijevaju otvoreno pisanje koda. Upravo takve igre pomažu kod procesa pažnje za vizualizacijom problema i vještina rješavanja istog.

Prema Li-u i Tsai-u (2013) povećavaju motivaciju korisnika to jest studenata i olakšavaju učenje složenijih programa. Samim time, motivira studente da razmišljaju i stvaraju strategije kako se rješavaju problemi u programskim jezicima, a opet im je to sve vizualno prikazano.

Također, bitno je napomenuti kako igre ne bi trebale predstavljati jednostavnost, ali niti preveliku kompleksnost kako ne bi došlo do preranog odustajanja koje dolazi zbog straha. Cilj im je upravo pomoći i na zanimljiv način prikazati studentima na koji način programi rade.

7.2.1. *CodeCombat*

CodeCombat video je igra za učenje programiranja također dostupna u web pregledniku u kombinaciji s Python-om.

Sastoji se od toga da kroz pisanje koda u programskom jeziku Python što je vidljivo na slici 15. Igrač prelazi dijelove igre te ono najvažnije oni izgledaju onako kako ih sam korisnik kreira pisanjem koda. Temelji se kao i većina video igri na izbjegavanju drugih igrača ili napadanja istih kako bi se riješio problem.

Sam početak igrice je dosta jednostavan i ne toliko složen te čak i oni koji se nikada nisu susreli sa programiranjem mogu shvatiti o čemu se radi i koji je cilj. Ipak, kako igra ide dalje i kako se stvaraju novi dijelovi tako oni počinju biti sve zahtjevniji. Time dolazi do toga da i samo pisanje koda počinje biti zahtjevnije. Pojavljuje se ponavljanja radnji, odnosno počinje se služiti petljama te postaje sve kompleksnije.

Uvelike pomaže u stvaranju samog sadržaja i učenja od osnovnih varijabli pa sve do naredbi i već spomenutih petlji. Igra pruža povratne informacije kako bi korisnik kasnije

razumio logiku koda koji je napisao. Korisnik sam bira jezik na kojem će temeljiti pisanje svog koda te je samim time svaki kod poseban i autentičan.

Kasnije služi kao jako dobar temelj za izgradnju drugih, težih i kompleksnijih aplikacija ili možda čak i stvaranje nekih grafičkih igri te je odličan način za ulazak u programiranje.

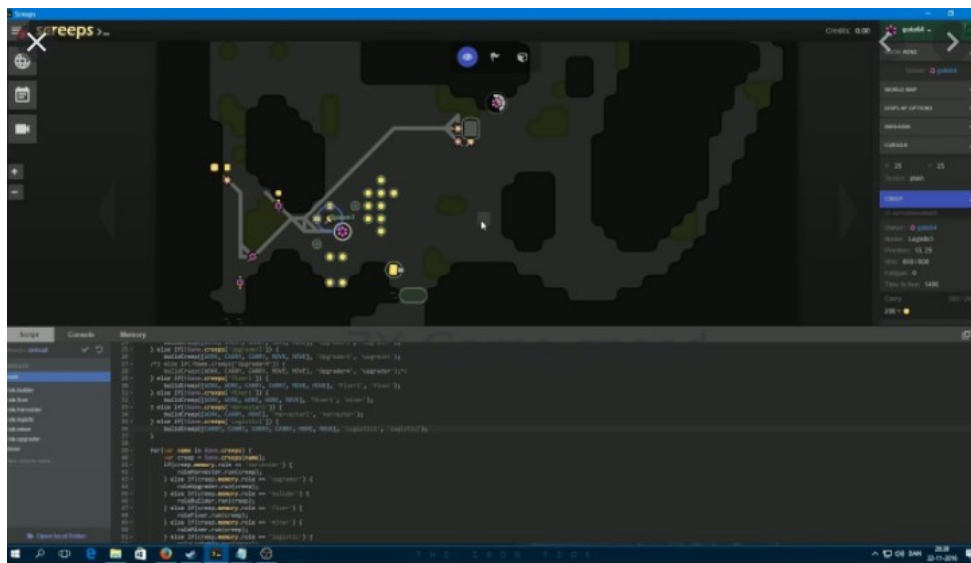


Slika 15 - Okruženje CodeCombat igre za studente (Common sense education, 2018)

7.2.2. Screeps

Screeps je strateška igra koja može služiti i kao igra za suradničko učenje jer se radi o igri s više igrača odjednom i događa se u stvarnom vremenu. Igra kao takva je ipak razvijenija za korisnike koji imaju vještine u programiranju. Sadrži sve što bi studenti koji dolaze iz na primjer srednjih škola informatičkog usmjerenja ili su prošli osnove programiranja trebali donekle znati. Kombinacija je petlji, funkcija i izravnog pisanja koda vidljivog na slici 16, što bi studentima koji se prvi put susreću sa programiranjem ipak mogao biti problem.

Bazira se na strategijama i osvajanju različitih teritorija i izaziva više vremena nego neke manje zahtjevnije igre. Ne koristi se toliko u edukacijske svrhe, ali ako je želja za dodatnim učenjem programiranja kroz igru velika ova igra može ispuniti takve kriterije pogotovo na višim godinama studija.



Slika 16 - Okruženje igre Screeps (Screeps tutorial, 2020)

7.2.3. Usporedna analiza igri za studente

IGRE ZA UČENJE PROGRAMIRANJA - studenti		
KRITERIJI	ALATI	
	CodeCombat	Screeps
Zainteresiranost korisnika za učenje	VRLO DOBRO	DOBRO
Jednostavnost učenja	DOBRO	DOVOLJNO
Dostupnost materijala i vodiča za učenje	DOBRO	DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	VRLO DOBRO	VRLO DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	DOBRO	DOBRO
Motiviranost korisnika za danjim učenjem	VRLO DOBRO	DOVOLJNO

Tablica 8 - Usporedna analiza igri za studente (Rad autorice)

U tablici 8 prikazana je usporedna analiza igri namijenjenih studentima prema zadanim kriterijima iz koje slijedi:

- **CodeCombat** – zainteresiranost studenta za pokušajem igranja ove igre namijenjene učenju programiranja se pokazala vrlo dobrom i to isključivo zbog vrlo dobrog pružanja informacija o potencijalnim problemima i pogreškama što samim time dobro pruža i različite mogućnosti i funkcionalnosti. Jednostavnost učenja je dobra, a dostupnost materijala također, iako se svodi na isključivo internet izvore i video tutorijale. Onime što nudi, dobro uvodi studente u programiranje te se takav način učenja kroz igre pokazao vrlo dobrim motiviranjem korisnika za daljnje učenje.
- **Screeps** – zainteresiranost korisnika za ovakvu igru je dobra jer se većinom radi o korisnicima koji već imaju dobro znanje programiranja. Dostupnost materijala i vodiča za učenje je dobra s obzirom da se svodi na internet izvore i dostupnost različitih materijala putem foruma i web stranica. Pružanje informacija o pogreškama i potencijalnim problemima je vrlo dobra, dok je pružanje različitih mogućnosti i funkcionalnosti dobro uz očekivanje da su korisnici već upoznati s programiranjem. Motiviranost korisnika za daljnjim učenjem je dovoljna i to isključivo iz razloga kompleksnosti igre koja zna prestrašiti korisnike kad se prvi put susretnu s njom.

7.3. Alati za stvaranje i uređivanje igri

Učenje programiranja kroz aplikacije za programiranje igri u grafičkom smislu jedne su od najpopularnijih među studentima. Ovo je već jedno od dijelova gdje se smatra da studenti znaju osnove programiranja u nekim od programskih jezika kao što su na primjer C++ ili Python, no čak i znanje Loga i Basic-a je u suradnji s ovakvim načinom kreiranja igri. Kod ovakvog programiranja igri, ne radi se o pisanju koda, već o onom većini zanimljivijeg dijela, a to je slaganje grafike, odnosno grafičkih ikona.

Ovakvi načini učenja programiranja nisu za nekoga tko se jednog dana želi baviti pisanjem koda, odnosno stvoriti nekakav program koji će biti rezultat njegovog rada jer kao što je već rečeno, radi se o slaganju (drag&drop). Dakle, za početnike kojima je programiranje u programskim jezicima više razine još prilično nepoznato ili nisu dobro upoznati s njim, dobar je uvod u petlje, grananja i različite funkcije koje se koriste u programiranju.

7.3.1. *GameMaker-Studio 1.4*

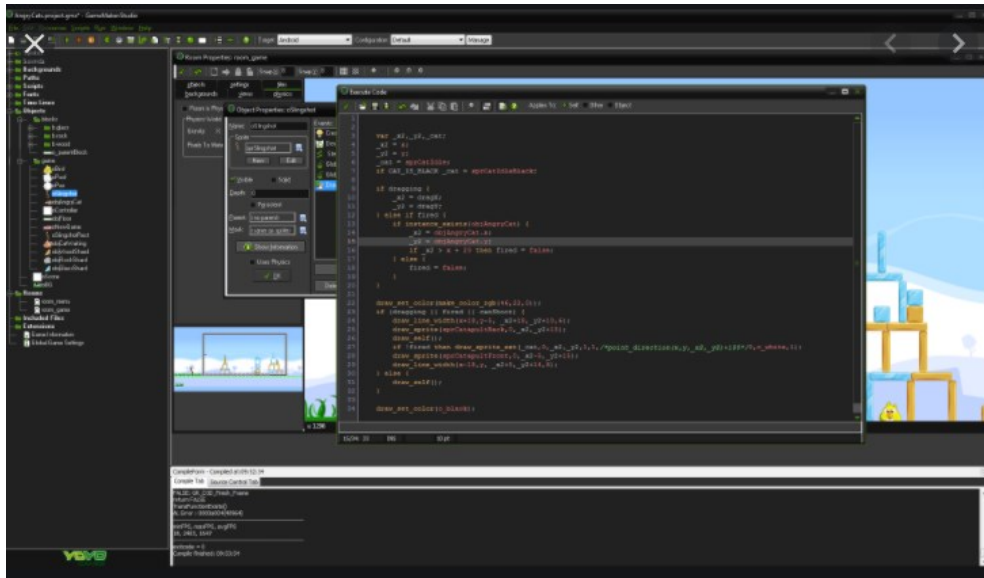
GameMaker-Studio inačice 1.4. je komercijalizirana aplikacija za dvodimenzionalno i trodimenzionalno stvaranje računalnih igrica.

Podržana je za Windows i Mac, ali postoje istraživanja vezana i za Android. Ipak, jedna je od aplikacija koja nije besplatna, ali kao i većina takvih aplikacija postoji 30 dana besplatnog korištenja nakon čega se plaća licenca i to u pozamašnom iznosu.

U inačici koja nudi besplatno korištenje od već navedenih 30 dana, korisnicima su dostupni svi resursi kojima se sama aplikacija koristi kao što su na primjer razni gotovi oblici ili kombinacija oblica i pozadine.

U ovoj aplikaciji za učenje programiranja igri postoje praktički sve naredbe koje se koriste i u programiranju u objektno-orientiranim programima poput petlji, funkcija, nizova i različitih načina sortiranja te pisanja izvornog koda što je vidljivo na slici 17.

Također, radi se o aplikaciji koja zahtjeva dobro znanje programiranja u kojoj se mogu koristiti i različiti objekti i strukture te se samim time i očekuje dobro ili vrlo dobro manevriranje u kodiranju.



Slika 17 - Okruženje Game-Maker Studio inačica 1.4 (Wraithkal, 2014)

7.3.2. Construct 2

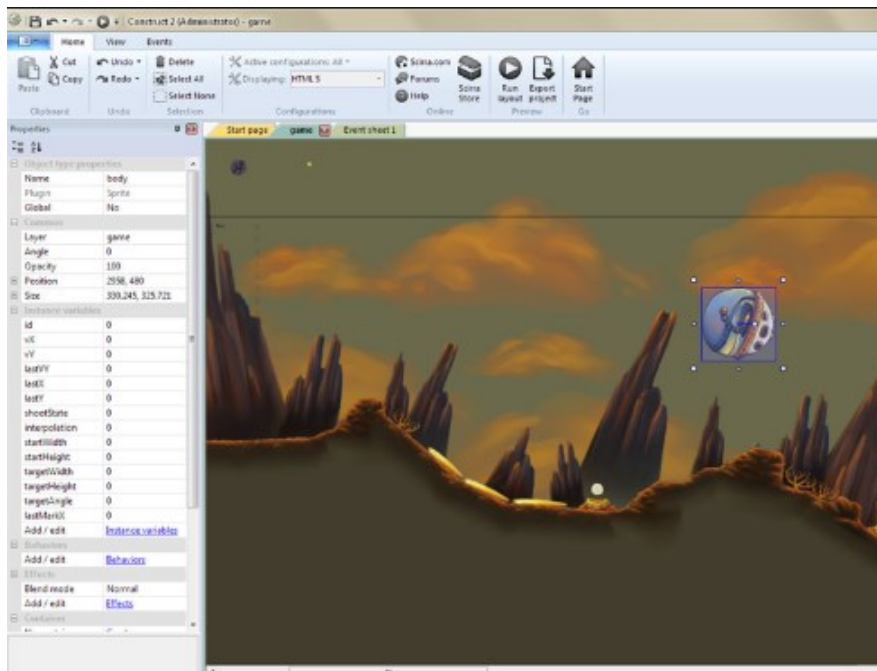
Također jedna od komercijalnih aplikacija za učenje programiranja u kreiranju igri. Ne zahtjeva nikakvo klasično kodiranje, nego kao i već spomenuta GameMaker-Studio 4.1. radi se od drag&drop aplikaciji.

Ipak, za početnike i studente koji se jednog dana žele baviti kreiranjem raznih video igri i to u grafičkom smislu, ali žele više programiranja, ovo nije loš odabir za kreiranje sadržaja i upoznavanjem sa tematikom te načinom upotrebe programa.

Kao i prethodnik, aplikacija nije besplatna i plaćaju se licence u nešto manjem cjenovnom rangu nego gore navedeno, ali i dalje dosta visoko. Ipak, postoji mogućnost kupovanja licence u poslovnom smislu što dosta sveučilišta u Americi koristi kako bi svojim studentima pružili i taj način kreiranja igrica.

Mnogi ga nazivaju i igrom za Photoshop-om za programiranje jer postoji velika količina alata za manevriranje sa fotografijama kao što je prikazano na slici 18.

Za ovo grafičko programiranje igri potrebno je dobro znanje programiranja kao takvog, pogotovo u kombinaciji s grafičkim znanjima za što se uglavnom prema literaturama i dostupnim izvorima odlučuju studenti različitih grafičkih fakulteta.



Slika 18 - Construct 2 okruženje (Scirra, n.d.)

7.3.3. Usporedna analiza alata za stvaranje i uređivanje igri

U tablici 9 prikazana je usporedna analiza alata za stvaranje i uređivanje prema zadanim kriterijima te je svakom dodijeljena ocjena zadovoljavanja kriterija.

ALATI ZA STVARANJE I UREĐIVANJE IGRI		
KRITERIJI	ALATI	
	GM-Studio 1.4	Construct 2
Zainteresiranost korisnika za učenje	DOBRO	VRLO DOBRO
Jednostavnost učenja	DOBRO	DOBRO
Dostupnost materijala i vodiča za učenje	DOBRO	DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	DOBRO	DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	VRLO DOBRO	VRLO DOBRO
Motiviranost korisnika za danjim učenjem	DOBRO	VRLO DOBRO

Tablica 9 - Usporedna analiza grafičkih igri (Rad autorice)

- **GameMaker-Studio 1.4.** – zainteresiranost korisnika za korištenjem ove aplikacije je dobra što objašnjava i dobru jednostavnost učenja i dostupnost materijala i vodiča za učenje koji se u globalu svode na internetske izvore i različite video tutorijale te forume. Pružanje informacija o pogreškama i potencijalnim problemima je dobra za razinu na kojoj aplikacija funkcionira i kakvo znanje programiranja unaprijed ona zahtjeva. Sadrži vrlo dobar spektar različitih mogućnosti i funkcioniranja u krugu programiranja te dobru motiviranost korisnika za daljnjim učenjem programiranja na temelju grafike.
- **Construct 2** – zainteresiranost korisnika za korištenjem ove aplikacije za grafičko programiranje igri se pokazala vrlo dobrom zbog jednostavnog učenja i pružanja različitih mogućnosti i funkcionalnosti. Dostupnost materijala i vodiča za učenje se kao i kod većine ovakvih alata nalazi u online bazama i različitim forumima, a postoji i veliki broj video tutorijala. Upravo zbog cijelog dojma kojeg ovakvo programiranje igri ostavlja, motiviranost korisnika za daljnjim učenjem programiranja u ovom alatu je vrlo dobra.

8. Učenje programiranja u višim programskim jezicima

Nakon prelaska određenog dijela alata koji su namijenjeni učenju programiranja poput onih namijenjenih za Web 2.0 sučelje pa preko suradničkog načina učenja programiranja, najjednostavnijih blog-a i wiki-ja pa sve do vizualnih alata za dijagrame tokova, alata za vizualizaciju te na kraju mini-jezika, učenja programiranja putem igri jedini logični slijed koji ima dodatne alate za lakše učenje programiranja su programi visoke razine, odnosno viši programski jezici.

Kroz dijagrame tokova učenici su naučili i vizualno si predložili kakav je tok izvršenja programa, kroz mini-jezike kreće zainteresiranost za programiranjem, a kroz učenje programiranja kroz igru motivacija i znatiželja sve više rastu.

Nakon što većina učenika, korisnika ili studenata shvati logiku programiranja i stvaranja te pisanja manjih kodova, prelaze na kompletno pisanje otvorenog koda kojeg ovi programski jezici nude.

Kako bi se tema skrenula na konačni odabir jezika u kojem će se sami program pisati, dobro je znati što ti programi sami po sebi moraju imati te koji su kriteriji bitni kako bi program zadovoljio ono što korisnik želi napisati i na kraju krajeva mogao raditi ono što je u konačnici ideja osobe koja piše program i bila.

Bitne stavke kod odabira programa su definitivno da program ima dovoljno literature koja je kvalitetna, na fakultetu da nastavnici i profesori imaju dobre i kvalitetne nastavne materijale. I kao bitna stavka da je jezik sam po sebi dopušten za korištenje u nastavnom kurikulumu, ali i prema različitim preferencijama samih nastavnika i profesora ima takozvano zeleno svjetlo. Također, program prije svega mora biti razumljiv studentima. Prednost kod početka programiranja je prije svega da su se studenti sreli prije toga sa bilo kakvim načinom programiranja.

Prilikom pisanja koda važno je da studenti ne gube motivaciju i samopouzdanje ako kod na prvo "probavanje" ne radi jer samim time očekivanja su velika, ali za njih kao što je rečeno treba puno strpljenja, truda, rada, volje ali i dobre organizacije.

U nastavku se donose alati koji služe za analizu koda te samim time pomažu korisnicima, odnosno studentima ili učenicima. Na različite načine alati pomažu, odnosno kroz različite sugestije kao kod na primjer pravilnog pisanja sintakse ili pronalaženja grešaka koje su učinili u samom pisanju koda. Upravo takvi alati uvelike pomažu korisnicima kako bi došli do cilja, a glavni cilj im je da zamišljeni program radi ono što su oni naumili. Naravno, to nije uvijek tako i dosta često se potkradu nekakve sitne greške za koje studenti uglavnom misle da su sitne, no nekad znaju stvarati stvarno velike probleme u daljnjem pisanju koda.

Ova skupina alata za učenje viših programskih jezika podijeljena je u dvije skupine programskih jezika:

- **Programske jezike opće namjene**
 - Python
 - Java
 - C++
- **Programske jezike prilagođene za rad na Web-u**
 - PHP
 - JavaScript

Kao alati za ovu analizu korišteni su:

- **Python** – *Prospector* i *Pylint*
- **Java** – *QAPlug* i *CheckStyle*
- **C++** - *Verificator* i *Cppcheck*
- **PHP** – *Zen Studio* i *Rapid PHP Editor*
- **JavaScript** – *Atom* i *Sublime Text*

9. Učenje programiranja u višim programskim jezicima opće namjene

Programski jezici opće namjene su oni programski jezici koji nisu namijenjeni samo i isključivo uskom području primjene nego je njima moguće rješavati veliki i širok raspon različitih zadataka. Uglavnom se radi o objektno-orijentiranom programiranju što je u principu postupak pisanja nekog programa sa upotrebom različitih skupova objekata tj. zasebnih programa koji su u mogućnosti obaviti različite zadatke, a razmjenjivati poruke u isto vrijeme.

Upravo takva vrsta programskih jezika neovisna je o samom računalu i operacijskom sustavu te internetu jer isti mogu raditi i van mreže.

Najbolji primjeri takvih programskih jezika su: Python, Java, C, C++, Ruby i dr.

9.1. Programski jezik Python

Van Rossum osamdesetih godina prošlog stoljeća razvio je programski jezik Python. Definiran je kao programski jezik opće namijene te je *high-level* (odnosno program više razine). Može se koristiti na dva načina i to kao sistemski jezik te kao skriptni jezik.

Glavna značajka i namjena programskog jezika je njegova laka čitljivost dostupna preko vizualnog rasporeda. Za pisanje koda se koristi engleski jezik što je vidljivo na slici 19, no kao i kod većine programskih jezika.

Bitne značajke:

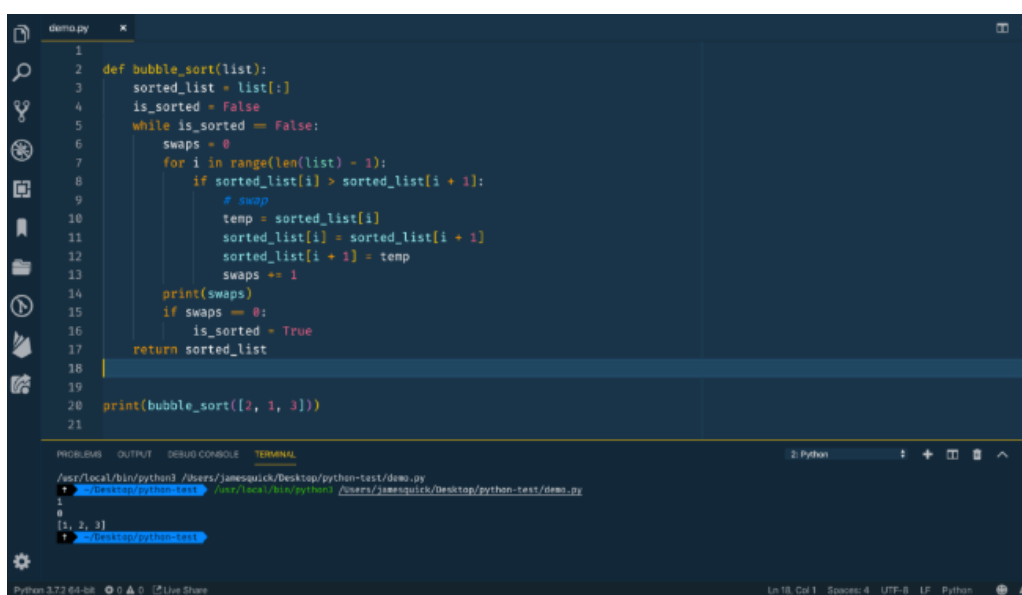
1. **Proširivost** – sadržava mogućnost dodavanja različitih modula koji programerima omogućuju prilagođavanje različitih alata za što bolju učinkovitost
2. **Jednostavnost** – kada se mislili na jednostavnost misli se na jednostavnost koda koja je jasno definirana i uočljiva
3. **Jednostavnost učenja** – kod ove vrste jednostavnosti smatra se da je Python jedan od najjednostavnijih početničkih programa za učenje programiranja jer koristi samo nekoliko ključnih riječi

4. **Prenosivost** – Python kao takav podržava veliki broj platformi koje sadrže isto sučelje

Ipak, unutar informacijsko-komunikacijskih zajednica postoji kritika i to ona da je Python vrlo spor programski jezik. Budući da je interpreterski jezik logično je da se kodovi u njemu kompajlaju sporije od nekih drugih programskih jezika poput na primjer C++-a.

Veliku sličnost pridodaju i Javi jer su oboje su slični jezici koji imaju gotovo iste načine podrške i samim time i jedan i drugi koriste samo jednu procesorsku jezgru.

Ipak, neovisno o svim vrlinama i manama, Python se danas smatra kao ključni i glavni programski jezik osobnog računala, a sve je zastupljeniji u najranijim susretima s programiranjem u osnovnim školama.



```
demo.py
1
2 def bubble_sort(list):
3     sorted_list = list[:]
4     is_sorted = False
5     while is_sorted == False:
6         swaps = 0
7         for i in range(len(list) - 1):
8             if sorted_list[i] > sorted_list[i + 1]:
9                 # swap
10                temp = sorted_list[i]
11                sorted_list[i] = sorted_list[i + 1]
12                sorted_list[i + 1] = temp
13                swaps += 1
14            print(swaps)
15            if swaps == 0:
16                is_sorted = True
17        return sorted_list
18
19
20 print(bubble_sort([2, 1, 3]))
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python

```
/usr/local/bin/python /Users/jamesquik/Desktop/python-test/demo.py
1
0
[1, 2, 3]
```

Python 3.7.2 64-bit Live Share Ln 18, Col 1 Spaced: 4 UTF-8 LF Python

Slika 19 - Programiranje u programskom jeziku Python (GeeksforGeeks, n.d.)

9.1.1. Prospector

Prospector je alat za analizu koda koji pruža informacije o pogreškama i potencijalnim problemima što je vidljivo na slici 20. Za razliku od drugih alata koje Python nudi, Prospector nudi nekoliko različitih profila koji su već u knjižicama koje se nalaze u projektima sučelja programskog jezika.

Veliku prednost ima što se tiče pronalaženja grešaka te također nudi i mogućnost ispunjavanja unaprijed te daje prijedloge za kod.

Namijenjen je studentima koji imaju prilično dobro znanje programiranja u programskom jeziku te na taj način daje i dodatnu motivaciju za daljnji rad.

```
> prospector --strictness low
Messages
=====

main.py
Line: 13
mccabe: MC0001 / run is too complex (17)

Check Information
=====
Started: 2015-04-11 15:59:47.759944
Finished: 2015-04-11 15:59:51.598176
Time Taken: 3.84 seconds
Formatter: grouped
Profiles: default, strictness_low, strictness_medium,
strictness_high, strictness_veryhigh, no_doc_warnings,
no_test_warnings, no_member_warnings
Strictness: low
Libraries Used:
Tools Run: dodgy, mccabe, pep8, profile-validator, pyflakes,
pylint
Messages Found: 1
```

Slika 20 – Prospector analiza koda (Github,, 2017)

9.1.2. Pylint

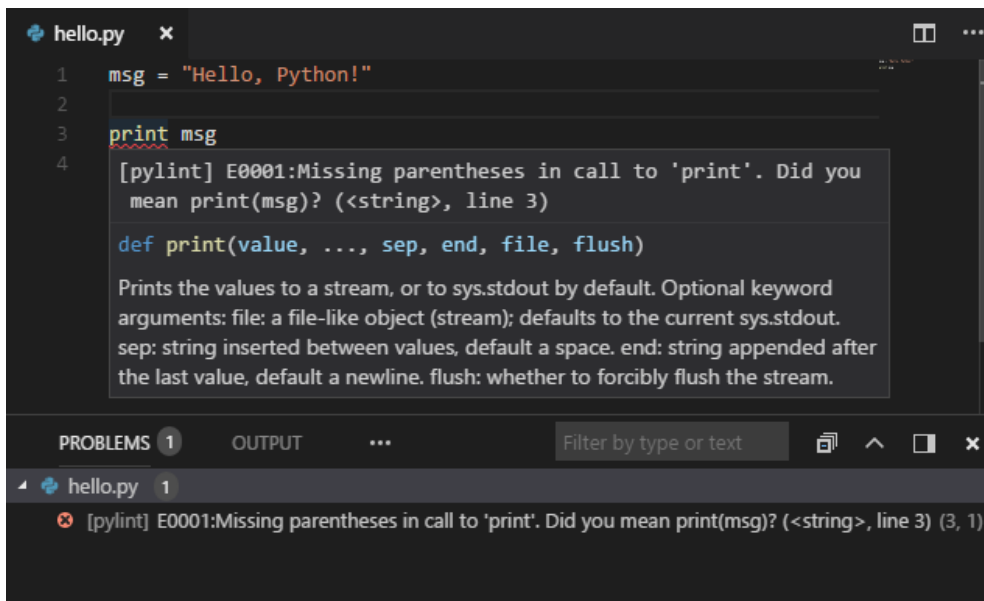
Pylint je alat za izvorni kod, provjeru programske pogreške i provjeru programskog koda napisanog u programskom jeziku Python. Pomaže u prevođenju osnovnih standarda kodiranja i nudi jednostavne prijedloge i sugestije za vrijeme pisanja koda te pokretanje komandnih linija.

Konfiguracija mu je prilično jednostavna te ima i posebne paragrame za kontrolu pogrešaka te pruža upozorenja za potencijalne probleme koda i samim time može konfigurirati vrlo velike i opsežne datoteke.

Nadalje, alat je kojim se provjerava duljina svakog retka, provjerava jesu li imena svih varijabli dobro napisana i jesu li u skladu sa standardima kodiranja programa u programskom jeziku Python što je vidljivo na slici 21.

Također, opremljen je i modulima koji omogućuju generiranje dijagrama tokova iz napisanog koda što mu je velika prednost za razliku od ostalih alata za pomoć u programiranju u višim programskim jezicima.

Besplatan je i može se koristiti kao samostalan alat, ali mu se mogu dodati i različita integrirana programska okruženja poput Eclipse ili Visual Studio Code.



The screenshot shows a code editor window with a file named 'hello.py'. The code contains the following lines:

```
1 msg = "Hello, Python!"
2
3 print msg
4
```

A tooltip is displayed over the 'print' function call on line 3, showing the following error message:

```
[pylint] E0001:Missing parentheses in call to 'print'. Did you mean print(msg)? (<string>, line 3)
```

Below the error message, the signature of the 'print' function is shown:

```
def print(value, ..., sep, end, file, flush)
```

Below the signature, a description of the function is provided:

```
Prints the values to a stream, or to sys.stdout by default. Optional keyword arguments: file: a file-like object (stream); defaults to the current sys.stdout. sep: string inserted between values, default a space. end: string appended after the last value, default a newline. flush: whether to forcibly flush the stream.
```

At the bottom of the editor, the 'PROBLEMS' panel is visible, showing the error message:

```
[pylint] E0001:Missing parentheses in call to 'print'. Did you mean print(msg)? (<string>, line 3) (3, 1)
```

Slika 21 – Korištenje alata Pylint (Visual Studio Code, n.d.)

9.1.3. Usporedna analiza alata programskog jezika Python

U tablici 10 prikazana je usporedna analiza alata za programski jezik Python. Dodijeljena je ocjena zadovoljavanja kriterija za svaki pojedini kriterij.

PROGRAM OPĆE NAMJENE - PYTHON		
KRITERIJI	ALATI	
	Prospector	Pylint
Zainteresiranost korisnika za učenje	VRLO DOBRO	VRLO DOBRO
Jednostavnost učenja	VRLO DOBRO	VRLO DOBRO
Dostupnost materijala i vodiča za učenje	VRLO DOBRO	VRLO DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	VRLO DOBRO	ODLIČNO
Pružanje različitih mogućnosti i funkcionalnosti	VRLO DOBRO	ODLIČNO
Motiviranost korisnika za daljnjim učenjem	VRLO DOBRO	VRLO DOBRO

Tablica 10 - Usporedna analiza alata programskog jezika Python (Rad autorice)

- **Prospector** – generalno, Prospector se pokazao kao odličan alat koji pomaže studentima i ostalim korisnicima u analizi koda prilikom učenja programiranja. Zainteresiranost korisnika za učenje je vrlo dobra jer je jedan od onih alata koji je jednostavan za učenje zbog same vrlo dobre dostupnosti materijala i vodiča za učenje koji su u većini pohranjeni u online oblicima te različitim video tutorialima. Prospector kao takav vrlo dobro pruža uvid o informacijama o pogreškama kao i nastajanju novih problema za samo vrijeme pisanja koda te pruža različite mogućnosti i funkcionalnosti. Uz sveopće dobre osobine ovog alata, motiviranost korisnika za daljnjim učenjem programiranja se pokazala vrlo dobrom.
- **Pylint** – također kao i njegov prethodnik se pokazao odličnim alatom koji pomaže u učenju programiranja. Velika zainteresiranost korisnika zbog same jednostavnosti primjene i učenja te online dostupnosti materijala i vodiča se pokazala vrlo dobrom. Ipak se za razliku od Prospectora pokazao za mrvicu boljim zbog boljeg pružanja informacija o pogreškama i problemima te pružanju različitih mogućnosti i funkcionalnosti kao što su na primjer izrada dijagrama toka. Također, motiviranost korisnika za daljnjim učenjem se pokazala vrlo dobrom.

9.2. Programski jezik Java

Java je zauzela veliku prednost nad drugim programskim jezicima koji su do tada postojali jer je pisanje programa u Javi moguće izvoditi bez bilo kakvih promjena u operacijskim sustavima za koje postoji *Java Virtual Machine*. Java danas podržava veliki broj aplikacija te posjeduje svoj vlastiti C# i .NET kao platformu, a sve to kao odgovor na svoj *open source*.

Većinu sintakse preuzima tj. nasljeđuje iz programskog jezika C++ ma da je za razliku od C++-a, Java ipak teži funkcionalni program. Programi odnosno kodovi koji se pišu u Javi uvijek se nalaze u klasama što je vidljivo na slici 22, tako da glavna i izvršna *main* funkcija moraju biti unutar klase.

Također, Java pruža visoku razinu sigurnosti jer je prije svega usmjerena ka mobilnoj vrsti uređaja odnosno pametnih telefona koji podatke u većini slučajeva dijele preko mreže te se samim time čini vrlo sigurnom kako bi se spriječilo ne dozvoljeno dijeljenje podataka prilikom korištenja samog programa.

```
1 package net.javastring.strings;
2
3 import java.util.Scanner;
4
5 public class JavaStringSwitchCase {
6
7     public static void main(String[] args) {
8
9         System.out.print("Please enter any programming language:\n");
10
11         Scanner scanner = new Scanner(System.in);
12
13         String input = scanner.next();
14
15         scanner.close();
16
17         switch (input) {
18             case "java":
19                 System.out.println("Java Current Version is 12.");
20                 break;
21             case "python":
22                 System.out.println("Python Current Version is 3.7");
23                 break;
24             case "rust":
25                 System.out.println("Rust Current Version is 1.34.1");
26                 break;
27             default:
28                 System.out.println("We don't have information about this pr
29         }
30     }
31 }
```

Slika 22 - Primjer pisanja otvorenog koda u programskom jeziku Java (*Javastring, n.d.*)

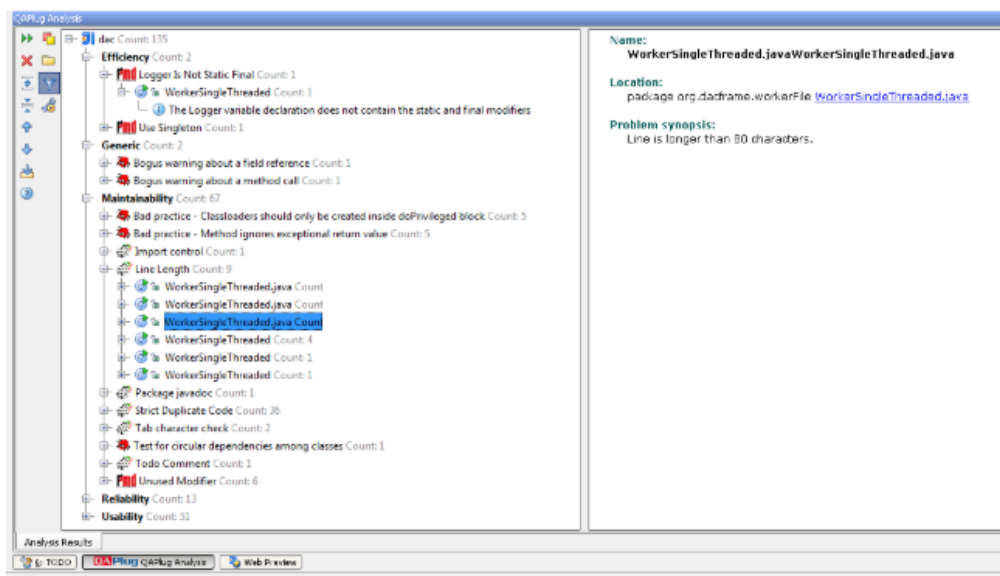
9.2.1. QAPlug

Alat QAPlug besplatan je alat tj. dodatak za programski jezik Java te prvenstveno pomaže u upravljanu kvalitete samog koda.

Korisnicima omogućuje različite mogućnosti i funkcije vidljive na slici 23, koje se mogu koristiti kao na primjer: statičke analizatora koda, analizatore koda koji provjeravaju više dijelova koda i pronalaze probleme i mogućnosti dizajna klasa u programu.

Također otkriva različite probleme napravljene tijekom dizajniranja samog koda, odnosno pisanja sintakse. Statički stabilizatori koda pronalaze neiskorištene, a imenovane varijable te objekte koji su stvoreni u pisanju, a nisu iskorišteni.

Postoji još jedan analizador koda u alatu, a to je pronalaženje potencijalnih grešaka za vrijeme pisanja koda ili neispravnost u radu samog programa.



Slika 23 - Primjer alata QAPlug u programskom jeziku Java (QAPlug tutorial, n.d.)

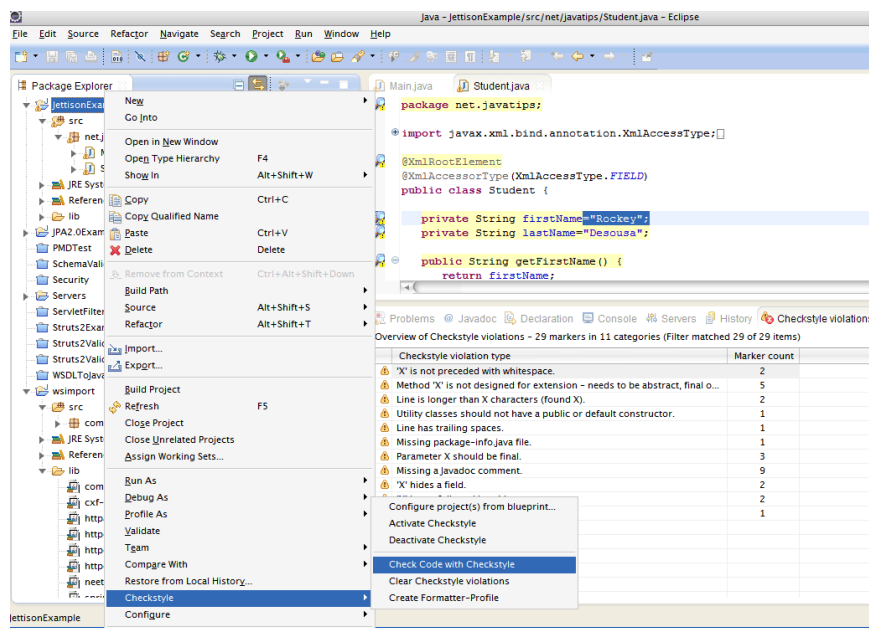
9.2.2. Checkstyle

Checkstyle je alat za statičku analizu koda razvijen 2001. godine s ciljem provjere pravila kodiranja izvornog koda što je vidljivo na slici 24. Trenutačna aktivna verzija ovog alata je inačica 8.34 koja se najčešće koristi i usmjerena je ka Javi 8.

Alat je koji generalno može pomoći oko usklađenosti koda te poboljšati njegovu čitljivost i kvalitetu. Same provjere su uglavnom usmjerene na samu prezentaciju koda što bi značilo da iste ne potvrđuju njegovu potpunost ili ispravnost.

Sadrži vlastiti skup dostupnih modula koji pružaju različite mogućnosti kao na primjer: *Javadoc* komentare za klase, atribute i metode, imenovanje atributa, broj funkcija, duljinu linije koda, prisutnost zaglavlja koji su obavezni, korištenje modifikatora, razmake između znakova i različita mjerenja složenosti.

Uglavnom koristi vlastite Java datoteke te se može pokretati unutar *Java Virtual Machine*, ali može se i integrirati u druga integrirana programska okruženja.



Slika 24 - Checkstyle analiza koda (Vogella, 2016)

9.2.3. Usporedna analiza alata programskog jezika Java

U tablici 11 prikazana je usporedna analiza alata za programski jezik Java. Dodijeljena je ocjena zadovoljavanja kriterija za svaki pojedini kriterij.

PROGRAM OPĆE NAMJENE - JAVA		
KRITERIJI	ALATI	
	QAPlug	Checkstyle
Zainteresiranost korisnika za učenje	VRLO DOBRO	DOBRO
Jednostavnost učenja	VRLO DOBRO	DOBRO
Dostupnost materijala i vodiča za učenje	VRLO DOBRO	VRLO DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	VRLO DOBRO	DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	VRLO DOBRO	DOBRO
Motiviranost korisnika za danjim učenjem	VRLO DOBRO	VRLO DOBRO

Tablica 11 - Usporedna analiza alata programskog jezika Java (Rad autorice)

- **QAPlug** – generalno se ovaj alat pokazao kao vrlo dobrim za učenje programiranja. Zainteresiranost korisnika koji se odluče za programiranje u programskom jeziku Java je vrlo dobra zbog jednostavnog učenja, ali i vrlo dobre dostupnosti materijala i vodiča za učenje koje se pružaju kroz različite internet izvore, ali i stručnu literaturu. Pružanje informacija o pogreškama i potencijalnim problemima kao i različite mogućnosti i funkcionalnosti poput provjere pisanja koda, provjere klasa i zasebnih dijelova koda su vrlo dobre. Samim time, QAPlug se pokazao vrlo dobrim u motiviranosti korisnika za daljnjim učenjem programiranja i ne odustajanja od učenja istog.
- **Checkstyle** – zainteresiranost korisnika kao i jednostavnost učenja za ovim alatom se pokazala dobrom isključivo zbog nešto manje dostupnog pružanja informacija o potencijalnim pogreškama i problemima te pružanju različitih mogućnosti i funkcionalnosti općenito zbog statičke analiziranja koda. Iako pruža velik spektar mogućnosti za analizu, za učenje programiranja je ipak nešto teži od prethodnog, no za korisnike koji se žele dodatno posvetiti i proširiti svoje znanje pokazao se vrlo dobrim za motiviranost za daljnjim učenjem.

9.3. Programski jezik C++

C++ je objektno-orientirani programski jezik opće namjene za pisanje izvornog koda što je vidljivo na slici 25. Ima veliku sličnost s programskim jezikom C, no ipak program sam za sebe. Prvobitni naziv mu je bio "*C s klasama*" jer je imao sva svojstva s klasama iz programskog jezika C. Sam napredak i poboljšanje programskog jezika dovele su do poboljšanja funkcija, operatora, višestrukog nasljeđivanja te predloške s i bez izmjena.

Prema Stoustrup (1994) postoje **pravila i različiti principi** kod samog dizajna programskog jezika C++ a to su:

1. C++ je dizajnirani jezik tako da bude jezik opće namjene te da je efikasan i prenosiv poput jezika C
2. C++ pruža različite mogućnosti za korištenje više stilova u programiranju, kako proceduralno programiranje tako i apstrakciju podataka, generičko programiranje i objektno-orientirano programiranje
3. C++ jezik je dizajniran upravo tako da bude kompatibilan sa programskim jezikom C i to u što većoj mogućoj razini
4. C++ treba izbjegavati implementaciju značajki koje nisu opće namjene ili su specifične za neku posebnu platformu
5. C++ je također dizajniran kako bi programerima ponudio više izbora u načinu korištenja samih značajki programa jer u većini slučajeva restrikcije prijete korisniku da radi ono kako je naumio
6. C++ bi trebao izbjegavati značajke koje njemu nisu potrebne
7. C++ je programski jezik koji je dizajniran tako da bude funkcionalan bez obzira na kompleksnost drugih programskih okruženja

```
averageOf1.cpp X
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      cout << "Basic arithmetic\n\n";
7
8      int ione;
9      int itwo;
10
11     cout << "Enter your first number: " << endl;
12     cin >> ione;
13     cout << "Enter your second number: " << endl;
14     cin >> itwo;
15
16
17     int iplus = ione + itwo;
18     int iminus = ione - itwo;
19     int imultiply = ione * itwo;
20     int idivide = ione / itwo;
21     int imodulus = ione % itwo;
22
23
24     cout << "Addition\t" << iplus << endl;
25     cout << "Subtraction\t" << iminus << endl;
```

Slika 25 - Primjer pisanja otvorenog koda u okruženju C++ (Tehnology now, n.d.)

Također, nije na odmet za spomenuti kako C++ kao programski jezik je prenosiv te je jako dobar izvor za veliki broj platformi. C++ ima jako veliki izbor već gotovih biblioteka te dopušta iznimke, a koristi također jako veliki broj aplikacija od grafike kao i sve matematičke logike u realnom vremenu.

C++ dobar je izbor na učenje programiranja u srednjim školama i fakultetima jer omogućuje dublje gledanje u svijet programiranja te kao takav daje temelj za još kompleksnije programe koje koriste programeri.

9.3.1. Verificator

Programiranje općenito u svakom višem programskom jeziku tako i u C++ prije svega zahtjeva razumijevanje i određene vještine kako bi se svladala programska problematika. Verificator prema definicijama je alat koji studentima pomaže u učenju programiranja, a nastavnicima u pripremanju zadataka namijenjenih studentima za vrijeme nastave.

Prema Radoševiću et. al. (2009) Verificator ima određene značajke koje su kao takve bitna karakteristika kako bi se programiranje olakšalo studentima:

1. **Autentičnost samog koda** – značilo bi da Verificator sprječava unošenje koda iz drugih izvora, odnosno vanjskih izvora
2. **Vremensko ograničenje** – korisnici, odnosno studenti ili učenici moraju u vremenu koje im je zadano završiti zadatak koji im je namijenjen. Nakon određenog vremena koje je zadano kao kraj vremena, program se može pokrenuti, ali se izmjene ne mogu raditi
3. **Personalizacija programa** – studenti imaju slobodu da unose vlastite podatke kao što su na primjer ime i prezime, te broj indeks-a ili nešto drugo, koji se zapisuju u sami kod, ali u obliku komentara čime se postiže da bi se kasnije moglo provjeriti je li program napisan u Verificatoru
4. **Stvaranje i usvajanje programskih navika** – Verificator sam po sebi zahtjeva provjeru koda kod svakih deset unosa koji su prikazani u semaforu u grafičkom obliku. Nakon deset unosa, Verifikator više ne dopušta pokretanje koda.

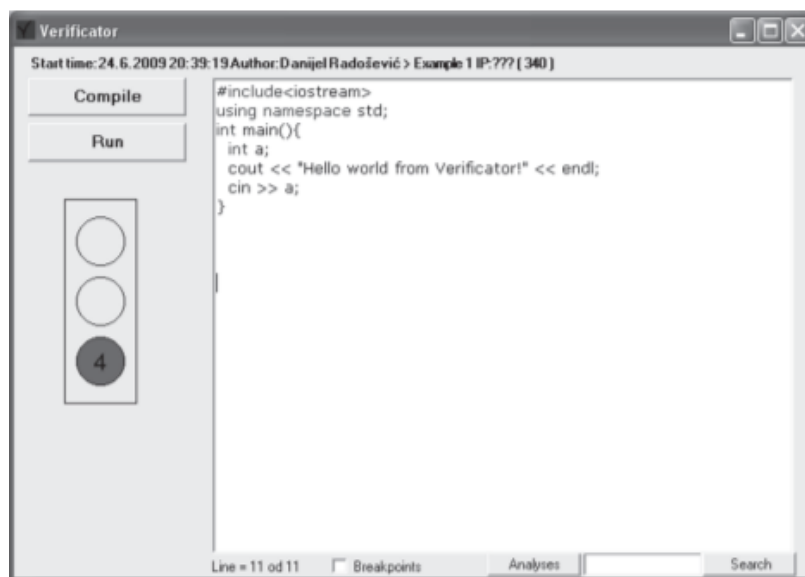
Glavne funkcije Verificatora su pomoć u sintaksi, kontrolne točke u razvoju programa te sama personalizacija programa. Personalizacija počinje tako da se student treba prijaviti nakon čega se podaci spremaju u autentičnu datoteku te se nakon toga u programskom jeziku pokazuju u obliku komentara.

Dobra stvar u korištenju Verificatora za edukacijske svrhe je ta da ne postoji mogućnost kopiranja programskog koda iz i u sami Verificator. Dakle, postoje kontrolni brojevi (uglavnom prvi redovi) koji će profesorima ili nastavnicima omogućiti da vide da li je programski kod prepisan ili ne.

Semafor koji se nalazi u sučelju, što je vidljivo na slici 26, ukazuje na dopuštene ili ne dopuštene radne. Zeleno pokazuje da se pisanje koda može nastaviti, žuto svjetlo znači da treba provjeriti pogreške, dok kad se pojavi crveno svjetlo niti samo kompajlanje nije dopušteno jer je veliki broj smetnji.

Nadalje, za vrijeme testiranja Verificator mijenja određene programske blokove te ih pokazuje kao postotak od ukupnog broja programskih blokova koji su dostupni. Upravo takav način ne dopušta korisnicima, odnosno u najčešćem slučaju studentima da pišu programski kod unaprijed, odnosno da na primjer pišu funkciju, a da je prije toga nisu pozvali ili da granaju program na petlje bez da imenuju varijable i sl.

Danas se Verificator uvelike gradi te se koristi među studentima na većini europskih sveučilišta. Priznat je u Bolonjskom nastavnom procesu, a prije svega omogućuje učenje pisanja koda i pomaže studentima u pobjedi straha od programiranja.



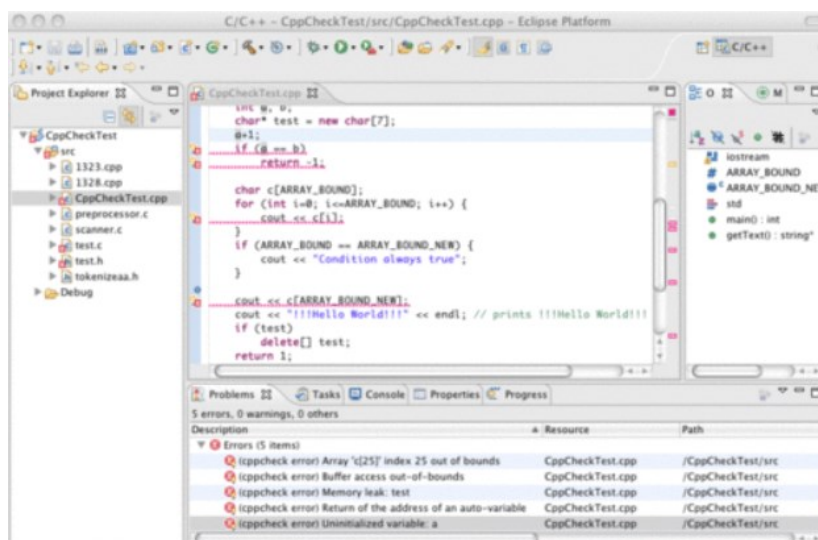
Slika 26 - Okruženje alata Verificator (Radošević D. et. al., 2009, str. 271)

9.3.2. Cppcheck

Cppcheck alat je za statičku analizu koda napisanog u programskom jeziku C++. Pruža mogućnosti poput otkrivanja različitih bugova, ali se u principu najviše usredotočuje na otkrivanje opasnih konstrukcija pisanja koda što je vidljivo na slici 27. Cilj mu je da upravo na taj način otkrije vrlo malo lažnih rezultata programiranja. Također, bitno je napomenuti kako je definiran na način da može analizirati u C++-u i C-u čak i ako imaju nestandardnu sintaksu odnosno onu koja nije poznata projektima koji su već ugrađeni.

Uz analizu otkrivanja bugova dostupno je i sučelje samog naredbenog retka kao i grafičko korisničko sučelje. Ono što je jedinstveno za razliku od drugih alata je osjetljivost same analize što bi generalno značilo kako će Cppcheck otkriti veći broj nepravilnosti tj. bugova nego na primjer neki drugi alat. Kao takav Cppcheck pronalazi probleme prije nego što se oni dogode, odnosno mogu analizirati kao što je prije rečeno izravnog koda (bez da se isti pokreće) te pronaći greške poput: preusmjerivanja null pokazivača, pretrpavanje memorije, pristup izvan granica, pogreške u varijablama i slično.

Cilj kompajlera je generirati izvršnu (binarnu) datoteku i prema zadanim postavkama se fokusirati na statičku analizu koda. *Open source* je te podržava različite dodatke poput: *CLion*, *Codacy*, *Eclipse*, *Visual Studio* i slično.



Slika 27 - Cppcheck analiza koda (Software Tasting Tools Guide, n.d.)

9.3.3. Usporedna analiza alata programskog jezika C++

U tablici 12 prikazana je usporedna analiza alata za programski jezik C++. Dodijeljena je ocjena zadovoljavanja kriterija za svaki pojedini kriterij.

PROGRAM OPĆE NAMJENE - C++		
KRITERIJI	ALATI	
	Verificator	Cppcheck
Zainteresiranost korisnika za učenje	ODLIČNO	VRLO DOBRO
Jednostavnost učenja	VRLO DOBRO	VRLO DOBRO
Dostupnost materijala i vodiča za učenje	VRLO DOBRO	VRLO DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	ODLIČNO	VRLO DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	VRLO DOBRO	DOBRO
Motiviranost korisnika za danjim učenjem	ODLIČNO	VRLO DOBRO

Tablica 12 - Usporedna analiza alata programskog jezika C++ (Rad autorice)

- **Verificator** – zainteresiranost korisnika za ovim alatom za učenje programiranja je odlična i to osobito zbog njegova odličnog pružanja informacija o pogreškama i potencijalnim problemima kao i vrlo dobrim različitim mogućnostima i funkcionalnostima s kojim dolazi. Prilično je jednostavan za učenje, naravno uzimajući u obzir da se koristi u višim programskim jezicima i najviše služi korisnicima koji imaju vrlo dobro znanje programiranja. Dostupnost materijala i vodiča za učenjem je također vrlo dobra i svodi se na veliki broj online vodiča, video tutorijala i stručne literature dostupne u bazama podataka sveučilišta. Samim time, motiviranost korisnika za daljnjim učenjem programiranja je odlična.
- **Cppcheck** – kao jedan od najpoznatijih statičkih analizatora koda u C++ zainteresiranost korisnika za korištenjem istog je vrlo dobra upravo zbog njegove jednostavnosti učenja i dostupnosti materijala i vodiča za učenje koji se također svode na online izdanja različitih članaka, video uradaka i znanstvenih radova. Pruža vrlo dobre informacije o pogreškama i potencijalnim problemima za vrijeme pisanja koda i velika mu je prednost što kod može provjeravati i bez njegova pokretanja. Ima dobre različite mogućnosti i funkcionalnosti što ha time čini vrlo dobrim za motiviranost korisnika za daljnjim učenjem programiranja.

10. Učenje programiranja u višim programskim jezicima za rad na Web-u

Web programiranje kao takvo sadrži veliki broj različitih mogućnosti i načina za modeliranje različitih stranica ili pak aplikacija.

Dakle, Web programiranje se odnosi na pisanje, označavanje i kodiranje koje je uključeno u sam razvoj što samim time uključuje različite sadržaje i skriptiranje klijenata te mrežnu sigurnost preko poslužitelja. Najčešći jezici koji se koriste u Web programiranju su: XML, HTML, JavaScript, Perl 5 i PHP.

Razlikuje se od programiranja u programima opće namijene prema samom znanju programiranja u aplikacijskom području i samoj tehnologiji baza podataka.

10.1. PHP

PHP je popularni skriptni jezik namijenjen programiranju na WEB-u. Smišljen je kao skriptni jezik tako da se može koristiti za prikupljanje različitih podataka iz drugih obrazaca, dinamičko generiranje sadržaja samih stranica, ali i za primanje i slanje različitih dodataka.

Koristi se često kao umetak u HTML i time se kreiraju web stranice, ali nije ograničen samo na HTML već može generirati i različite fotografije ili na primjer PDF datoteke.

Ono što ga čini drugačijim od drugih je definitivno njegova mogućnost podrške za različite baze podataka te je time prilično jednostavno nadograditi aplikacije. Također, osim samih bazi podataka PHP ima i mogućnosti komuniciranja sa različitim internet servisima na primjer IMAP, POP3, HTTP i slično, što je vidljivo na slici 28.

Znanje PHP-a danas je iznimno cijenjeno kao znanje programiranja jer je nebitno žele li se korisnici baviti samo izradom jednostavnih stranica ili programiranjem velikih aplikacija.

Samim time i popularnosti PHP-a kao takvog svakim danom broj web stranica i aplikacija napisanih upravo u ovom skriptnom jeziku svakodnevno raste te se broj kvalitetnih programera nadograđuje i raste.

```
1 <?php
2 /*
3  * Extract the 'id' value from the query string.
4  * We're filtering to ensure it matches the expected integer number we expect.
5  */
6 $id = filter_input(INPUT_GET, 'id', FILTER_SANITIZE_NUMBER_INT);
7 // Filter the 'title' value, which can contain special characters (HTML
8 // entities and the like).
9 $title = filter_input(INPUT_GET, 'title', FILTER_SANITIZE_SPECIAL_CHARS);
10 // Build a simple URL with some good (matching) and bad values.
11 $goodURL = baseURL() . '?' . http_build_query(array('id' => 5276, 'title' => "Les Misérables"));
12 $badURL = baseURL() . '?' . http_build_query(array('id' => "not a number", 'title' => "Les Misérables"));
13 function baseURL() {
14     return (@$_SERVER['HTTPS'] ? "https://" : "http://") .
15           $_SERVER["SERVER_NAME"] .
16           ($_SERVER['SERVER_PORT'] != 80 ? ":" .
17           $_SERVER['SERVER_PORT'] : null) .
18           $_SERVER['SCRIPT_NAME'];
19 }
20 ?>
21 <!-- Our basic HTML output -->
22 <p>
23 Your searched book <?= $title ? "is titled $title" : "has no title" ?> and
24 has <?= $id ? "the ID $id." : "an invalid ID." ?>
25 </p>
26 <a href="<?= baseURL() ?>">Home</a><br/>
27 <a href="<?= $goodURL ?>">Good Link</a><br/>
28 <a href="<?= $badURL ?>">Bad Link</a><br/>
```

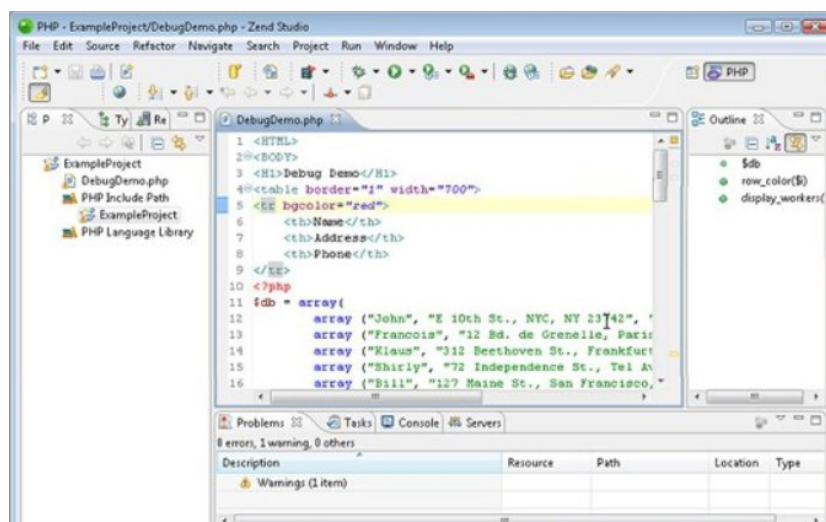
Slika 28 - PHP programiranje (Coding Dojo, 2015)

10.1.1. Zend Studio

Zend Studio je komercijalno integrirano razvojno okruženje (*IDE*) razvijeno za pomoć pri učenju PHP-a. Usko je povezan sa Zend poslužiteljem i njihovim unaprijed integriranim i testiranim paketom PHP aplikacija.

Jednostavan je za korištenje i samim time omogućuje korisnicima odnosno programerima da u vrlo kratkom vremenu postave kompletno PHP okruženje i samim time ubrzaju analizu programa i problema te svojim karakteristikama brzo otkriva probleme tijekom testiranja što je vidljivo na slici 29.

Posjeduje veliki broj dodataka koji služe za pronalaženje grešaka u pisanju koda, provjerava linije koda samim time i uvelike pomaže programerima u kreiranju kvalitetnog PHP koda.



Slika 29 - Zend Studio okruženje (Malavida, n.d.)

10.1.2. Rapid PHP Editor

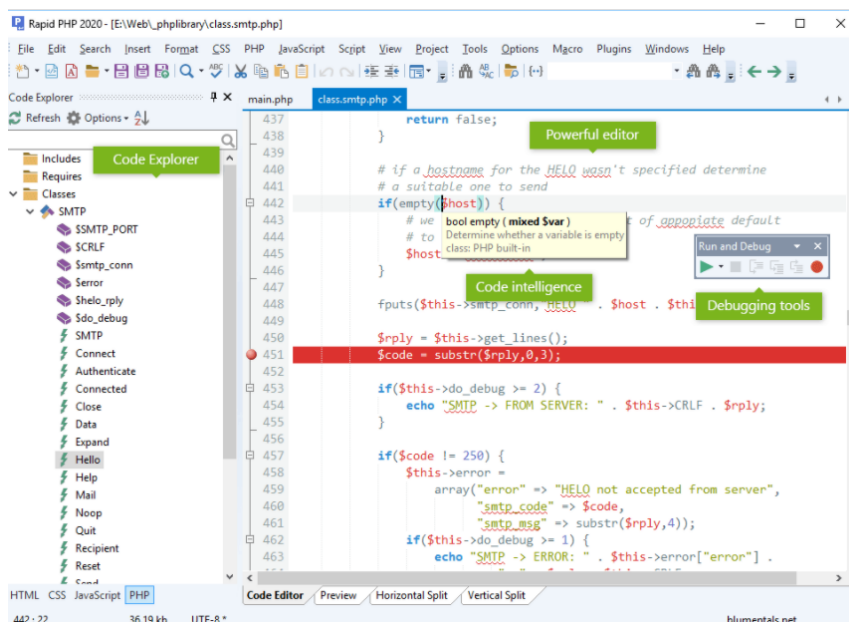
Rapid PHP Editor je snažni PHP uređivač koda za Windows koji se služi kombinacijom značajki potpunog PHP integriranog programskog okruženja s brzinom Notepad-a.

Rapid PHP je jedan od najcjelovitijih softvera za kodiranje, ne samo PHP-a nego i HTML-a, CSS-a i JavaScript-a te mnogih drugih jezika koji su namijenjeni razvoju. Alat je kojim se provjerava valjanost pisanja koda i samim time otklanjanjem pogrešaka. Pruža iznimnu navigaciju i formatiranje samog koda.

Uključuje funkcije poput automatskog dovršavanja, provjere same sintakse i posebnog programa za ispravljanje pogrešaka i mnogih drugih alata kojima se služi za razvoj brzog PHP-a.

Veliki je istraživač te ima moćno pretraživanje i zamjenu u samom kodu, istraživanje velikog broja bazi podataka, različite isječke iz kodova te provjeru valjanosti koda i pogrešaka kao što je vidljivo na slici 30.

Brz je i jak alat za učenje programiranja jer svim svojim karakteristikama uvelike pomaže u cilju učenja i napredovanja u programiranju.



Slika 30 - Rapid PHP Editor okruženje (Blumentals – rapid PHP, n.d.)

10.1.2. Usporedna analiza alata za PHP

U tablici 13 prikazana je usporedna analiza alata za učenje PHP-a. Dodijeljena je ocjena zadovoljavanja kriterija za svaki pojedini kriterij.

PRILAGOĐENI PROGRAMI ZA RAD NA WEBU - PHP		
KRITERIJI	ALATI	
	Zend Studio	Rapid PHP Editor
Zainteresiranost korisnika za učenje	DOBRO	ODLIČNO
Jednostavnost učenja	DOBRO	VRLO DOBRO
Dostupnost materijala i vodiča za učenje	DOBRO	VRLO DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	DOBRO	ODLIČNO
Pružanje različitih mogućnosti i funkcionalnosti	VRLO DOBRO	ODLIČNO
Motiviranost korisnika za danjim učenjem	DOBRO	ODLIČNO

Tablica 13 - Usporedna analiza alata za PHP (Rad autorice)

- **Zend Studio** – zainteresiranost korisnika za ovim komercijalnim alatom se pokazalo dobrim isključivo zbog zadovoljavajućih kriterija poput pružanja informacija o pogreškama i potencijalnim problemima i pružanju manjeg spektra različitih mogućnosti i funkcionalnosti. Dostupnost materijala i vodiča za učenje se svodi na internetske izvore te je motiviranost korisnika za daljnjim korištenjem i učenjem PHP-a preko ovog alata dala dobar učinak.
- **Rapid PHP Editor** – zainteresiranost korisnika za ovim brzim i jednostavnim PHP alatom se pokazala odličnom i to isključivo zbog odlične količine pružanja informacija o pogreškama i potencijalnim problemima s kojima se korisnik može susresti za vrijeme pisanja koda. Odličan je i u smislu pružanja različitih mogućnosti i funkcionalnosti koje nudi te mu je i vrlo dobra dostupnost materijala i vodiča za učenje koji se protežu kroz stručne znanstvene radove i literature kao i internet izvore, forume i video tutorijale. Samim time motiviranost korisnika za daljnjim učenjem je odlična.

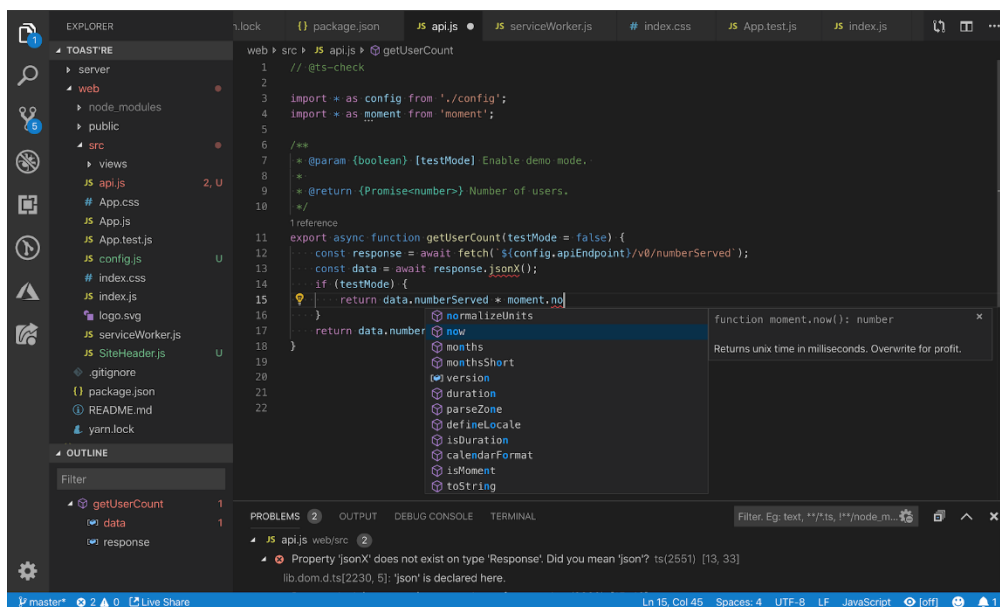
10.2. JavaScript

JavaScript je jedan od trenutno najpopularnijih skriptnih programskih jezika koji se izvršava preko preglednika.

Napravljen je sa ciljem da bude sličan samoj Javi zbog lakšeg korištenja, ali on za razliku od Jave nije objektno-orientiran programski jezik već se temelji na prototipu i samim time se tu značajno odvađa od samog programskog jezika Java.

Uz korištenje HTML-a, JavaScript je najpopularnija i jedna od osnovnih tehnologija kojima se služi Web. Omogućava različite web stranice te ih većina web lokacija koristi.

Kao programski jezik on podržava veliki broj stilova programiranja te posjeduje sučelja aplikacijskog programiranja. Također, podržava veliki broj različitih sučelja kao na primjer Visual Studio Code što je vidljivo na slici 31.



Slika 31 - JavaScript okruženje (Codeburst, 2018)

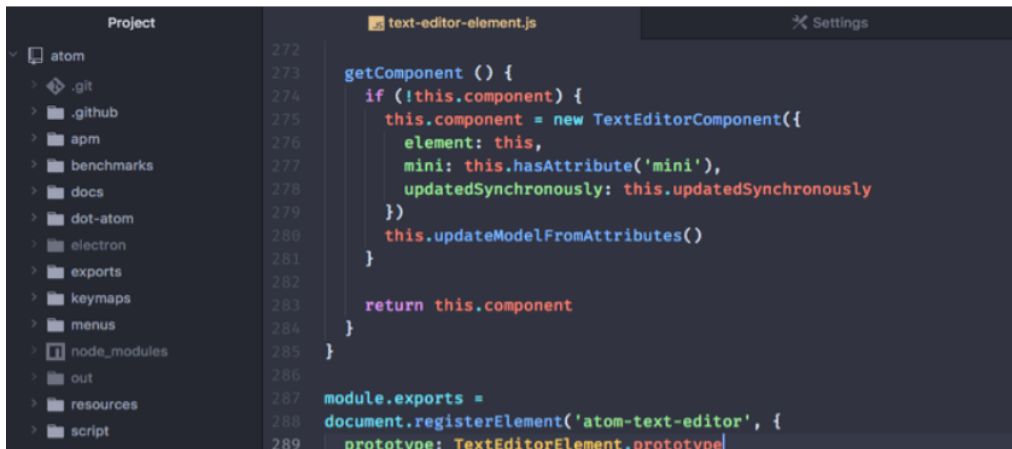
10.2.1. Atom

Atom je otvoreni uređivač teksta i izvornog koda, što je vidljivo na slici 32, što ga čini jednim od najpopularnijih uređivača za programiranje na Web-u, odnosno provjeru koda napisanog u programskom jeziku JavaScript.

Pruža i omogućuje korisnicima instaliranje različitih paketa i mogućnosti za uređivanje koda, automatske ispune za vrijeme pisanja koda, popravak pogrešaka i samim time uvelike pomaže u učenju programiranja pisanja koda u JavaScript-u.

Također, dostupan je za jako puno drugih programskih jezika, ne samo JavaScript, a nudi i bezbroj drugih dodataka koji se mogu instalirati unutar samog sučelja te koristiti za vrijeme pisanja koda.

Ono što ovaj alat radi je provjera izvornog koda, odnosno sam program nije potrebno pokrenuti. Samim time zainteresiranost za njim raste i to ne samo u smislu alata za provjeravanje koda u JavaScriptu nego i u drugim programskim jezicima namijenjenima programiranju na Web-u, na primjer HTML-u i CSS-u.



```
Project
└─ atom
├─ .git
├─ .github
├─ apm
├─ benchmarks
├─ docs
├─ dot-atom
├─ electron
├─ exports
├─ keymaps
├─ menus
├─ node_modules
├─ out
├─ resources
└─ script

text-editor-element.js
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289

getComponent () {
  if (!this.component) {
    this.component = new TextEditorComponent({
      element: this,
      mini: this.hasAttribute('mini'),
      updatedSynchronously: this.updatedSynchronously
    })
    this.updateModelFromAttributes()
  }
  return this.component
}

module.exports =
document.registerElement('atom-text-editor', {
  prototype: TextEditorElement.prototype
})
```

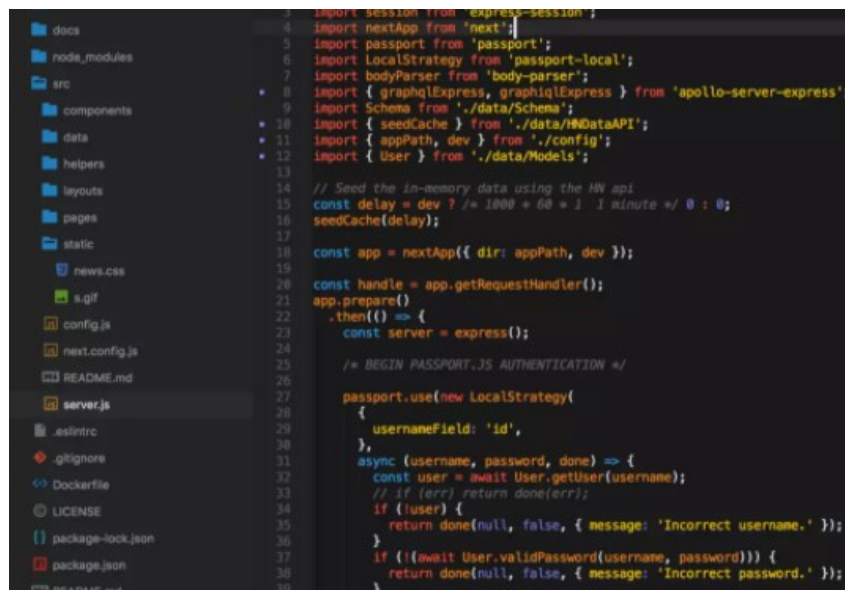
Slika 32 - Atom text editor (FOSSMint, 2017)

10.2.2. Sublime Text

Sublime Text zajednički je uređivač izvornih kodova (ne moraju se pokrenuti) za više različitih platformi s Python programskim sučeljem, no danas podržava mnoge programske jezike kako opće namjene tako i one kojima se vrši programiranje preko preglednika.

Sadrži veliki broj **funkcija i različitih mogućnosti** što je vidljivo na slici 33, koje korisnici mogu dodavati kao što su na primjer:

- Brze navigacije do datoteka i simbola ili pak linija samog koda
- Koristi zanimljivu paletu naredbi koje se koriste prilagodljivo
- Istodobno se mogu koristiti različite promjene na više različitih označenih područja
- Dostupan je na više platformi (*Windows, Linux, macOS*)
- Dostupan na više svjetskih jezika



```
1 import session from 'express-session';
2 import nextApp from 'next';
3 import passport from 'passport';
4 import LocalStrategy from 'passport-local';
5 import bodyParser from 'body-parser';
6 import { graphqlExpress, graphqlExpress } from 'apollo-server-express';
7 import Schema from './data/Schema';
8 import { seedCache } from './data/seedCache';
9 import { appPath, dev } from './config';
10 import { User } from './data/Models';
11
12 // Seed the in-memory data using the HW api
13 const delay = dev ? /* 1000 * 60 = 1 minute */ 0 : 0;
14 seedCache(delay);
15
16 const app = nextApp({ dir: appPath, dev });
17
18 const handle = app.getRequestHandler();
19 app.prepare().then(() => {
20   const server = express();
21
22   /* BEGIN PASSPORT.JS AUTHENTICATION */
23   passport.use(new LocalStrategy(
24     { usernameField: 'id',
25       async (username, password, done) => {
26         const user = await User.getUser(username);
27         // If (err) return done(err);
28         if (user) {
29           return done(null, false, { message: 'Incorrect username.' });
30         }
31         if (!(await User.validatePassword(username, password))) {
32           return done(null, false, { message: 'Incorrect password.' });
33         }
34       }
35     )
36   );
37 }
```

Slika 33 - Sublime Text u JavaScript-u (Sitepoint, n.d.)

10.2.3. Usporedna analiza alata za JavaScript

U tablici 14 prikazana je usporedna analiza alata za JavaScript. Dodijeljena je ocjena zadovoljavanja kriterija za svaki pojedini kriterij.

PRILAGOĐENI PROGRAMI ZA RAD NA WEBU - JAVASCRIPT		
KRITERIJI	ALATI	
	Atom	Sublime Text
Zainteresiranost korisnika za učenje	VRLO DOBRO	DOBRO
Jednostavnost učenja	VRLO DOBRO	DOBRO
Dostupnost materijala i vodiča za učenje	ODLIČNO	VRLO DOBRO
Pružanje informacija o pogreškama i potencijalnim problemima	VRLO DOBRO	VRLO DOBRO
Pružanje različitih mogućnosti i funkcionalnosti	VRLO DOBRO	VRLO DOBRO
Motiviranost korisnika za danjim učenjem	VRLO DOBRO	DOBRO

Tablica 14 - Usporedna analiza alata za JavaScript (Rad autorice)

- **Atom** – zainteresiranost korisnika za ovom vrstom alata za učenje se pokazala vrlo dobrom i to zbog vrlo dobre jednostavnosti učenja. Dostupnost materijala i vodiča za učenjem se pokazala odličnom u online smislu tj. dostupnosti literature, video tutorijala i različitih znanstvenih radova i specijaliziranih foruma. Pružanje informacija o pogreškama i potencijalnim problemima kao i različitih mogućnosti i funkcionalnosti su se pokazali kao vrlo dobri u motiviranosti korisnika za danjim učenjem programiranja i korištenjem ovog alata ne samo i isključivo u JavaScript-u nego i drugim programskim jezicima.
- **Sublime Text** – nakon dosta popularnog editora Atom-a, ovaj alat se pokazao dobrim što se odnosi na jednostavnost učenja i zainteresiranosti korisnika za iste. Dostupnost materijala i vodiča za učenje kao i pružanje informacija i potencijalnim problemima kao i pružanje različitih mogućnosti i funkcionalnosti su se pokazale vrlo dobrima. Ipak, nasuprot Atom-u za ovaj alat se odluči manji broj korisnika tako da je preko njega motiviranost korisnika za daljnjim učenjem nešto manja nego kod Atom-a, i uzimajući u obzir da je on samo tekstualni editor, a ne IDE i dalje daje zadovoljavajuće dobre ocjene.

11. Zaključne usporedne analize alata za učenje programiranja

Zaključne analize su također podijeljene u skupine i to:

- Web 2.0 okruženje alati za učenje programiranja
- Vizualni alati za učenje programiranja
- Igre za učenje programiranja
- Viši programski jezici za učenje programiranja

Svakoj skupini su dodijeljene ocjene od 1 do 5 ovisno o unaprijed napravljenoj analizi koja je korištenjem literature navedene u popisu literature i Internet izvora rad autorice i njezinog vlastitog mišljenja.

Sa zbrojem svih ocjena za svaki pojedinačni alat napravljen je prosjek konačne ocjene i donesen zaključak koji od alata daje najbolje mogućnosti u svojoj skupini.

11.1. Web 2.0 okruženje zaključna analiza

Zaključna analiza Web 2.0 i alata za suradničko učenje programiranje prikazana je u tablici 15. Svakom alatu dodijeljene su ocjene zadovoljavanja kriterija prema učinjenoj analizi te je za svaki alat napravljen prosjek ocjena.

ZAKLJUČNA ANALIZA - WEB 2.0. I SURADNIČKO UČENJE PROGRAMIRANJA				
KRITERIJI	PaizaCloud IDE	Codeanywhere	Blog	Wiki
Zainteresiranost korisnika za učenje	3	5	5	3
Jednostavnost učenja	4	4	4	5
Dostupnost materijala i vodiča za učenje	3	3	3	2
Pružanje informacija o pogreškama i potencijalnim problemima	4	4	2	2
Pružanje različitih mogućnosti i funkcionalnosti	3	5	4	2
Motiviranost korisnika za danjim učenjem	3	5	4	3
PROSJEK	3,33	4,33	3,66	2,83

Tablica 15 - Zaključna analiza Web 2.0 alata (Rad autorice)

Prema napravljenoj analizi i dodijeljenim ocjenama Web 2.0 alata za učenje programiranja najboljim se pokazalo okruženje **Codeanywhere** s prosjekom zadovoljavanja kriterija 4,33 što generalno dosta visoko zadovoljava sve zadane kriterije za ovu analizu.

Prema suradničkom programiranju u usporedbi bloga i wikija boljim se svakako pokazao **blog** sa prosjekom zadovoljavanja kriterija 3,66 i same zainteresiranosti korisnika i pružanja različitih mogućnosti i funkcionalnosti.

11.2. Vizualni alati zaključna analiza

Zaključna analiza vizualnih alata za učenje programiranja prikazana je u tablici 16. Svakom alatu dodijeljene su ocjene zadovoljavanja kriterija prema učinjenoj analizi te je za svaki alat napravljen prosjek ocjena.

ZAKLJUČNA ANALIZA - VIZUALNI ALATI ZA UČENJE							
KRITERIJI	RAPTOR	FlowGorithm	Gliffy	ViSA	PITON	MSWLogo	Guido van Robot
Zainteresiranost korisnika za učenje	5	4	5	5	5	5	5
Jednostavnost učenja	4	3	3	4	4	5	4
Dostupnost materijala i vodiča za učenje	4	4	5	3	3	5	4
Pružanje informacija o pogreškama i potencijalnim problemima	3	3	4	3	3	3	4
Pružanje različitih mogućnosti i funkcionalnosti	4	4	4	3	4	3	4
Motiviranost korisnika za danjim učenjem	5	3	5	4	4	3	4
PROSJEK	4,16	3,5	4,33	3,66	3,83	4	4,16

Tablica 16 - Zaključna analiza vizualnih alata (Rad autorice)

Prema napravljenoj analizi i dodijeljenim ocjenama u sklopu alata za izradu dijagrama toka bolju ocjenu prema kriterijima je postigao **RAPTOR** zbog svoje jednostavnosti učenja i same motiviranosti korisnika za danjim učenjem.

U sklopu alata za pomoć vizualizacije algoritma i alat ViSA i alat PITON imaju vrlo malo razlike, a ona je jedina u pružanju različitih mogućnosti i funkcionalnosti tako da je **PITON** s prosjekom ocjene 3,83 u ovoj analizi pogodniji alat za učenje programiranja.

U sklopu alata mini-jezika kao boljim se pokazao **Guido van Robot** i to zbog većeg pružanja informacija o pogreškama i potencijalnim problemima te različitih mogućnosti i funkcionalnosti te time s prosjekom ocjene 4,16 stvorio vodstvo i veću motiviranost korisnika za danjim učenjem programiranja.

11.3. Igre za učenje programiranja zaključna analiza

Zaključna analiza igri za učenje programiranja prikazana je u tablici 17. Svakom alatu, odnosno igri su dodijeljene ocjene zadovoljavanja kriterija prema učinjenoj analizi te je za svaki alat napravljen prosjek ocjena.

ZAKLJUČNA ANALIZA - IGRE ZA UČENJE PROGRAMIRANJA						
KRITERIJI	Daisy the dinosaur	AC: Coding game	CodeCombat	Screeeps	GM Studio 1.4	Construct 2
Zainteresiranost korisnika za učenje	3	5	4	3	3	4
Jednostavnost učenja	5	4	3	2	3	3
Dostupnost materijala i vodiča za učenje	3	3	3	3	3	3
Pružanje informacija o pogreškama i potencijalnim problemima	2	3	4	4	3	3
Pružanje različitih mogućnosti i funkcionalnosti	3	4	3	3	4	4
Motiviranost korisnika za danjim učenjem	3	4	4	2	3	4
PROSJEK	3,17	3,83	3,5	2,83	3,17	3,5

Tablica 17 - Zaključna analiza igri za učenje (Rad autorice)

Prema napravljenoj analizi i dodijeljenim ocjenama u kategoriji igri za djecu i mlađe skupine boljom se pokazao **Algorithm City** s prosjekom ocjena 3,83 i to zbog boljeg zadovoljavanja kriterija kao zainteresiranosti korisnika, pružanja informacija o pogreškama i problemima.

Kod igri za studente namijenjenih učenju programiranja definitivno se boljim pokazao **CodeCombat** s prosjekom zadovoljavanja kriterija 3,5 i to uključujući zainteresiranost studenata, jednostavnosti učenja, ali i motiviranosti za danjim učenjem.

U kategoriji alata za stvaranje i uređivanje igri, boljim se pokazao **Construct 2** s prosjekom ocjene 3,5 i to isključivo zbog veće motiviranosti korisnika za danjim učenjem programiranja, ali i zainteresiranosti samih korisnika.

11.4. Viši programski jezici zaključna analiza

Zaključna analiza alata namijenjenih višim programskim jezicima opće namjene prikazana je u tablici 18. Svakom alatu dodijeljene su ocjene zadovoljavanja kriterija prema učinjenoj analizi te je za svaki alat napravljen prosjek ocjena.

ZAKLJUČNA ANALIZA - VIŠI PROGRAMSKI JEZICI (OPĆI)						
KRITERIJI	Prospector	Pylint	QAPlug	CheckStyle	Verificator	Cppcheck
Zainteresiranost korisnika za učenje	4	4	4	3	5	4
Jednostavnost učenja	4	4	4	3	4	4
Dostupnost materijala i vodiča za učenje	4	4	4	4	4	4
Pružanje informacija o pogreškama i potencijalnim problemima	4	5	4	3	5	4
Pružanje različitih mogućnosti i funkcionalnosti	4	5	4	3	4	3
Motiviranost korisnika za danjim učenjem	4	4	4	4	5	4
PROSJEK	4	4,33	4	3,33	4,5	3,83

Tablica 18 - Zaključna analiza viših programskih jezika opće namjene (Rad autorice)

Prema napravljenoj analizi i dodijeljenim ocjenama boljim alatom za pomoć u programiranju kod programskog jezika Python se pokazao **Pylint** s prosjekom ocjene zadovoljavanja kriterija 4,33 i to zbog većeg pružanja informacija o pogreškama i potencijalnim problemima te pružanja različitih mogućnosti i funkcionalnosti.

Alat **QAPlug** za pomoć u programiranju kod programskog jezika Java se s prosjekom ocjena zadovoljavanja kriterija 4 pokazao bolji i to zbog svog jednostavnog učenja te pružanja informacija i potencijalnih problema te više različitih mogućnosti i funkcionalnosti.

Verificator kao alat za pomoć pri učenju programiranja u programskom jeziku C++ s prosjekom ocjene zadovoljavanja kriterija 4,5 se pokazao odličnim alatom i to prvenstveno zbog velike zainteresiranosti i motiviranosti korisnika, ali i svih ostalih kriterija koje odlično ili vrlo dobro zadovoljava.

Zaključna analiza alata namijenjenih višim programskim jezicima prilagođenim za rad na Web-u prikazana je u tablici 19. Svakom alatu dodijeljene su ocjene zadovoljavanja kriterija prema učinjenoj analizi te je za svaki alat napravljen prosjek ocjena.

ZAKLJUČNA ANALIZA - VIŠI PROGRAMSKI JEZICI (PRILAGOĐENI ZA RAD NA WEBU)				
KRITERIJI	Zend Studio	Rapid PHP Editor	Atom	Sublime Text
Zainteresiranost korisnika za učenje	3	5	4	3
Jednostavnost učenja	3	4	4	3
Dostupnost materijala i vodiča za učenje	3	4	5	4
Pružanje informacija o pogreškama i potencijalnim problemima	3	5	4	4
Pružanje različitih mogućnosti i funkcionalnosti	4	5	4	4
Motiviranost korisnika za danjim učenjem	3	5	4	3
PROSJEK	3,17	4,67	4,17	3,5

Tablica 19 - Zaključna analiza viših programskih jezika namijenjeni Web-u (Rad autorice)

Prema napravljenoj analizi i dodijeljenim ocjenama u kategoriji alata namijenjenih učenju programiranja PHP-a boljim se pokazao **Rapid PHP Editor** i to po svim ocjenama zadovoljavanja kriterija s prosjekom 4,67. Mogućnosti koje ovaj alat pruža za poboljšano i jednostavno učenje programiranja su ga učinili odličnim alatom za motiviranost korisnika za danjim učenjem programiranja.

U kategoriji alata namijenjenih učenju JavaScript-a boljim se pokazao **Atom** s prosjekom ocjene zadovoljavanja kriterija 4,17 i to zbog veće zainteresiranosti korisnika za učenje, jednostavnosti učenja, dostupnosti materijala te motiviranosti korisnika za danjim učenjem programiranja.

12. Zaključak

Unatrag nekoliko godina tehnologija je postala jedna od glavnih tema u svijetu. Za njezin razvoj i razvitak zaduženi su ljudi diljem svijeta. Prvenstveno, svojim znanjima i vještinama, a nakon toga upornošću i željom za napretkom. Danas, gdje god da se okrenemo ili pogledamo, tehnologija je oko nas. Veliki uspjesi u svim područjima, od biotehnologija do mehanike, elektrotehnike pa čak i poučavanja. Razvojnim programima koji danas postoje dostupno je mnoštvo toga. Od učenja na daljinu, do kompletnih stvari koje dovode do cilja zbog velikog truda, rada i upornosti. Kada bi pogledali oko sebe i samo svoj život, sve je na neki svoj način "programirano", no ipak, o programiranju kao takvom ne priča se toliko često.

Većina studenata se sa programiranjem prvi puta sretne na fakultetima i to uglavnom onim okrenuti ka tehničari, informatici i prirodnim znanostima kao takvima, a nastava informatike u osnovnim i srednjim školama svedena je prema nacionalnom kurikulumu na vrlo mali značaj programiranja kao takvog. Upravo je to, kako je i u Uvodu rada navedeno, bila najveća motivacija za pisanje upravo ovakve teme.

Postoji bezbroj načina kako se programiranje može prikazati na jako zabavan i interesantan način, ne samo studentima, nego i mlađim populacijama, ali i ljudima koji su svoj put okrenuli ka stvaranju programa kojim će se jednog dana koristiti veći broj ljudi.

Upravo alati koji su namijenjeni učenju mogu biti jako velika pomoć kod samog susretanja s programiranjem, jer na prvu ono zna biti dosta zastrašujuće, no kroz igru se također može doći do prvog bitnog zadatka kod učenja programiranja, a to je logički način razmišljanja kao jedno od najboljih karakteristika dobrog programera. Krenuvši od jednostavnih igri namijenjenih djeci, pa sve do igri za malo starije generacija do igri za studente, sve je to jako dobar način za ulazak u svijet nečega velikoga kao što je programiranje.

Znanjem i stjecanjem programske logike u igrama i kroz igru, napredak iz dana u dan može biti sve veći te vodi do onih viših programskih, odnosno objektno-orientiranih programa koji zahtijevaju i pisanje otvorenog koda te uvode direktno u veliki svijet programiranja kao takvoga te pružaju mogućnost napretka, stvaranja aplikacija,

tehnologije i svega onoga što je svijet dovelo do ove razine, a znanjem i umijećem dovodi do još većih i značajnijih znanstvenih napredaka.

Iako reforme u smislu školstva i nastave informatike u RH svake godine uvode nešto novo, razina za programiranjem je i dalje dosta malena, a velika pažnja se pridodaje samo osnovnim alatima i radu na računalu. Ipak, kroz godine je napredovalo dovoljno da se informatika kao predmet uvede već i niže razrede osnovnih škola kao obavezan predmet što uvelike donosi poboljšanju i reforme i nastavnih planova.

Upravo na taj način i uvođenjem informatike u razrednu nastavu svakako se povećava broj učenika zainteresiranih za informatiku i programiranje kao takvo, te se znanja i vještine za istim sve više razvijaju ka boljemu.

Analizom alata u ovom završnom radu prikazano je kako od igrica pa sve do složenih programa postoji jednostavan način za učenjem programiranja, na osnovu truda i uloženog vremena čitatelja koji se želi baviti programiranjem. Ipak, koliko tehnologija napreduje, svjesnost o napretku programskih jezika i samih alata je velika te su promjene moguće iz dana u dan. Sami sukus teme daje jasni prikaz u to kako programiranje može postati zabavno i interesantno većoj skupini ljudi, među kojima se ipak nađe netko tko bi nakon čitanja ovog rada usmjerio svoje interese ka programiranju i naišao na bitne podatke koji mu uvelike mogu pomoći kod učenja.

13. Literatura

PREDAVANJA, MEĐUNARODNE RECENZIJE I ZNANSTVENI ČLANCI

1. Škorić, I.; Pein, B.; Orehovački, T. Selecting the Most Appropriate Web IDE for Learning Programming Using AHP // *Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics* str. 982-987. Dostupno na: <https://www.bib.irb.hr/809869>
[pristupljeno: rujan, 2020]
2. Škorić, I.; Orehovački, T.; Ivašić Kos, M. *Learning to Code and Collaborate in a Web Environment* // Proceedings of the AHFE 2018 International Conference on Human Factors in Training, Education, and Learning Sciences, Advances in Intelligent Systems and Computing (AISC, volume 785) str. 54-56. Dostupno na: <https://www.bib.irb.hr/929413>
[pristupljeno: rujan, 2020.]
3. Sinožić, S.; Orehovački, T. Using Analytic Hierarchy Process to Select the Most Appropriate Tool for Learning Programming // *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics* / str. 600-605 Dostupno na: <https://www.bib.irb.hr/933834>
[pristupljeno: rujan, 2020.]
4. Orehovački, T.; Babić, S. Inspecting Quality of Games Designed for Learning Programming // *Proceedings of the 2nd International Conference on Learning and Collaboration Technologies (LCT 2015), Held as Part of the 17th International Conference on Human-Computer Interaction (HCI International 2015), Lecture Notes in Computer Science (LNCS, volume 9192)* / Zaphiris, Panayiotis Cham: Springer International Publishing Switzerland, 2015. str. 620-631. Dostupno na: <https://www.bib.irb.hr/753603?rad=753603>
[pristupljeno: rujan, 2020.]
5. Radošević, D.; Orehovački, T.; Lovrenčić, A. Verificator: Educational Tool for Learning Programming, (2009), 2; str. 261-280. Dostupno na: <https://www.bib.irb.hr/429295>
[pristupljeno: srpanj, 2020.]

6. Radošević, D.; Orehovački, T.; Lovrenčić, A. New Approaches and Tools in Teaching Programming // *Proceedings of the 20th Central European Conference on Information and Intelligent Systems* / Varaždin: Faculty of Organization and Informatics, 2009. str. 49-57. Dostupno na: <https://www.bib.irb.hr/427643>
[pristupljeno: rujan, 2020.]
7. Radošević, D.; Orehovački, T.; Stapić, Z. Automatic On-line Generation of Student's Exercises in Teaching Programming // *Proceedings of the 21st Central European Conference on Information and Intelligent Systems* / Varaždin: Faculty of Organization and Informatics, 2010. str. 87-93. Dostupno na: <https://www.bib.irb.hr/476823?rad=476823>
[pristupljeno: rujan, 2020.]
8. Radošević, D.; Orehovački, T. An Analysis of Novice Compilation Behavior using Verificator // *Proceedings of the 33rd International Conference on Information Technology Interfaces* / Zagreb: University Computing Centre, University of Zagreb, 2011. str. 325-330. Dostupno na: <https://www.bib.irb.hr/508878?rad=508878>
[pristupljeno: rujan, 2020.]
9. Đanić, M.; Radošević, D.; Orehovački, T. Evaluation of Student Programming Assignments in Online Environments // *Proceedings of the 22nd Central European Conference on Information and Intelligent Systems* / Varaždin: Faculty of Organization and Informatics, 2011. str. 111-116. Dostupno na: <https://www.bib.irb.hr/518061?rad=518061>
[pristupljeno: rujan, 2020.]
10. Reif, I.; Orehovački, T. ViSA: Visualization of Sorting Algorithms // *Proceedings of the 35th International Convention on Information and Communication Technology, Electronics and Microelectronics* / Rijeka: Croatian Society for Information and Communication Technology, Electronics and Microelectronics, 2012. str. 1146-1151. Dostupno na: <https://www.bib.irb.hr/566948>
[pristupljeno: rujan, 2020.]

KVALIFIKACIJSKI RADOVI

1. Škorić I., Alati za suradničko učenje programiranja, Sveučilište u Rijeci, (2010). Dostupno na:
http://www.inf.uniri.hr/files/studiji/poslijediplomski/kvalifikacijski/Skoric_Igor_kv_alifikacijski_rad.pdf [rujan, 2020.]

KNJIGE

1. Kovačević Ž. (2004.), *C++ analiza i primjena*, Školska knjiga, Zagreb
2. Kirriemuir, J., McFarlane, A. (2004.) *Literature Review in Games and Learning*. Bristol, Futurelab
3. Kusalić D. (2010.), *Napredno programiranje i algoritmi u C-u i C++-u*; Element, Zagreb
4. Šribar, J., Motik, B. (2003.), *Demistificirani C++*; Element, Zagreb
5. Dillenbourg P. (1999), *What do you mean by collaborative learning?* Oxford: Elsevier
6. Dillenbourg P. (1999), *Collaborative learning: Cognitive and Computational Approaches*; Oxford, Elsevier

INTERNET IZVORI:

1. *BMC Med Educ, A new generation of web-basted tools for virtual collaborative education* (2006) Dostupno na:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1564136/> [rujan, 2020]
2. *BLOODSHEDSOFTWARE, Dev-C++: Providing Free Software to the Internet community* (2015) Dostupno na: <https://www.bloodshed.net/devcpp.html> [rujan, 2020]
3. *Chudy, L., DZONE Sonar and QAPlug integration* (2011) Dostupno na: <https://dzone.com/articles/sonar-and-qaplug-integration> [rujan, 2020]
4. *{Code} {Hero} ACADEMY*. Dostupno na: <https://codeheroacademy.com/> [rujan, 2020]

5. *A Vary Basic Introduction to MSW Logo Programming* (2003) Dostupno na: <https://silo.tips/download/a-very-basic-introduction-to-msw-logo-programming> [rujan, 2020]
6. *SOFTONIC – Very simple Logo programming environment* (2019) Dostupno na: <https://mswlogo.en.softonic.com/> [rujan, 2020]
7. *What is a Blog? The Definition od Blog, Blogging, and Bloger* (2018) Dostupno na: <https://firstsiteguide.com/what-is-blog/> [rujan, 2020]
8. *O'Reilly – What Is Web 2.0 – Desing Patterns and Buiness for the Next Generation of Software* (2005) Dostupno na: <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html> [rujan, 2020]
9. *Flowgorithm*. Dostupno na: <http://www.flowgorithm.org/> [rujan, 2020]
10. *Python Code Analysis – Prospector*. Dostupno na: <http://www.bloodshed.net/devcpp.html> [rujan, 2020]
11. *Učenje programiranja grafičkim programiranjem igara* (2017) Dostupno na: <https://pogledkrozprozor.wordpress.com/tag/gamemaker-studio-1-4/> [rujan, 2020]
12. *Made with Construct*. Dostupno na: <https://www.construct.net/en/make-games/> [rujan, 2020]
13. *Python is powerful...and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open*. Dostupno na: <https://www.python.org/about/> [rujan, 2020]
14. *Prospector – Release 0.4* (2020) Dostupno na: <https://buildmedia.readthedocs.org/media/pdf/prospect/latest/prospect.pdf> [rujan, 2020]
15. *Pylint. Start your Python code!* Dostupno na: <https://www.pylint.org/> [rujan, 2020]
16. *The Java Language Specification – Java SE 8 Edition* (2015) Dostupno na: <https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf> [rujan, 2020]
17. *QAPlug quality assurance plugin*. Dostupno na: <https://qaplug.com/> [rujan, 2020]
18. *Using Static Analysis to inprove your Java Coding* (2020) Dostupno na: <https://www.eviltester.com/post/java/using-static-analysis-for-java.html> [rujan, 2020]

19. *Learn C++ Programming*. Dostupno na: <https://www.programiz.com/cpp-programming> [rujan, 2020]
20. *Teaching tools* (2020) Dostupno na: <https://gpml.foi.hr/laboratory/index.php?id=teaching-tools> [rujan, 2020]
21. *Cppcheck – A tool for static C/C++ code analysis*. Dostupno na: <http://cppcheck.sourceforge.net/> [rujan, 2020]
22. *PHP Tutorial*. Dostupno na: <https://www.w3schools.com/php/DEFAULT.asp> [rujan, 2020]
23. *Zend Studio PHP IDE*. Dostupno na: <https://www.zend.com/products/zend-studio> [rujan, 2020]
24. *Blumentals – Rapidphp – Lightweight PHP editor and PHP IDE*. Dostupno na: <https://www.rapidphpeditor.com/> [rujan, 2020]
25. *MDN web docs – What is JavaScript?* Dostupno na: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript [rujan, 2020]
26. *JavaScript language support in Atom*. Dostupno na: <https://atom.io/packages/language-javascript> [rujan, 2020]
27. *Atom IDE*. Dostupno na: <https://ide.atom.io/> [rujan, 2020]
28. *Turn Sublime Text 3 into a JavaScript IDE* (2018) Dostupno na: <https://css-tricks.com/turn-sublime-text-3-into-a-javascript-ide/> [rujan, 2020]
29. *PHP Example Code To Stay on Track* (2015) Dostupno na: <https://www.codingdojo.com/blog/php-example-code> [rujan, 2020]
30. *JavaScript in Visual Studio Code*. Dostupno na: <https://code.visualstudio.com/docs/languages/javascript> [rujan, 2020]
31. *Checkstyle*. Dostupno na: <http://www.testingtoolsguide.net/tools/checkstyle/> [rujan, 2020]
32. *Python code analysis using Prospector* (2015) Dostupno na: <http://0x7df.github.io/python-code-analysis-using-prospector.html> [rujan, 2020]
33. *Using the Checkstyle Eclipse plug-in – Tutorial* (2016) Dostupno na: <https://www.vogella.com/tutorials/Checkstyle/article.html> [rujan, 2020]
34. *Funcional Programming in JavaScript* (2018) Dostupno na: <https://codeburst.io/functional-programming-in-javascript-e57e7e28c0e5> [rujan, 2020]
35. *Gliffy* (2014) Dostupno na: <https://e-laboratorij.carnet.hr/gliffy/> [rujan, 2020]

14. Popis slika

Slika 1 - PaizaCloud okruženje	10
Slika 2 - Codeanywhere okruženje	11
Slika 3 - Uređivanje HTML-a u okruženju Wordpress	14
Slika 4 - Wikipedia okruženje	15
Slika 5 - Primjer dijagrama toka	18
Slika 6 - Izrada dijagrama toka u okruženju RAPTOR	19
Slika 7 - Izrada dijagrama toka u okruženju FlowGorithm	20
Slika 8 - Izrada dijagrama toka u okruženju Gliffy	21
Slika 9 - ViSA sortiranje	25
Slika 10 - Vizualizacija u alatu PITON	26
Slika 11 - MSWLogo okruženje	29
Slika 12 - Guido van Robot okruženje	30
Slika 13 - Daisy the dinosaur okruženje	33
Slika 14 - Algorithm City:Coding game okruženje	34
Slika 15 - Okruženje CodeCombat igre za studente	37
Slika 16 - Okruženje igre Screeps	38
Slika 17 - Okruženje Game-Maker Studio inačica 1.4	41
Slika 18 - Construct 2 okruženje	42
Slika 19 - Programiranje u programskom jeziku Python	47
Slika 20 - Prospector analiza koda	48
Slika 21 - Korištenje alata Pylint	49
Slika 22 - Primjer pisanja otvorenog koda u programskom jeziku Java	51
Slika 23 - Primjer alata QAPlug u programskom jeziku Java	52
Slika 24 - Checkstyle analiza koda	53
Slika 25 - Primjer pisanja otvorenog koda u okruženju C++	56
Slika 26 - Okruženje alata Verificator	58
Slika 27 - Cppcheck analiza koda	59
Slika 28 - PHP programiranje	62
Slika 29 - Zend Studio okruženje	63
Slika 30 - Rapid PHP Editor okruženje	64
Slika 31 - JavaScript okruženje	66
Slika 32 - Atom text editor	67
Slika 33 - Sumlme Text u JavaScript-u	68

15. Popis tablica

Tablica 1 - Ocjene kriterija	7
Tablica 2 - Usporedna analiza Web 2.0 alata	12
Tablica 3 - Usporedna analiza Web 2.0 alata za suradničko učenja	16
Tablica 4 - Usporedna analiza alata za učenje kroz dijagram toka	22
Tablica 5 - Usporedna analiza vizualizacijskih alata	27
Tablica 6 - Usporedna analiza mini-jezika	31
Tablica 7 - Usporedna analiza igri za djecu	35
Tablica 8 - Usporedna analiza igri za studente	39
Tablica 9 - Usporedna analiza grafičkih igri	43
Tablica 10 - Usporedna analiza alata programskog jezika Python	50
Tablica 11 - Usporedna analiza alata programskog jezika Java	54
Tablica 12 - Usporedna analiza alata programskog jezika C++	60
Tablica 13 - Usporedna analiza alata za PHP	65
Tablica 14 - Usporedna analiza alata za JavaScript	69
Tablica 15 - Zaključna analiza Web 2.0 alata	71
Tablica 16 - Zaključna analiza vizualnih alata	72
Tablica 17 - Zaključna analiza igri za učenje	73
Tablica 18 - Zaključna analiza viših programskih jezika opće namjene	74
Tablica 19 - Zaključna analiza viših programskih jezika namijenjeni Web-u	75

Sažetak

Prilikom bilo kakve usporedne analize, pa tako i ove za alate namijenjene učenju programiranja potreban je velik broj informacija potrebnih kako bi se kroz kriterije alati koji su korišteni mogli staviti jedan uz drugoga. Veliki fokus su prvenstveno bile internetske stranice, različiti forumi i blogovi na kojima se može naći veliki broj korisnih savjeta koje od alata koristiti kako bi pomogli, a koji su to alati koji možda neće biti dovoljni ili dovoljno jasni određenom broju korisnika. Ipak, u velikoj većini, forumi, blogovi nisu pokazali veliku pomoć oko izrade same analize, ali su u velikoj mjeri pomogli sa barem malim uvodom u ono što koji od alata generalno radi.

Okrenuti istraživanju brojnih znanstvenih članaka, članaka sa konferencija te već napisanih diplomskih i završnih radova te korištenjem stručne terminologije i literature počeo je dolaziti sve veći broj korisnih činjenica kako bi se sama analiza mogla provesti.

Kriteriji su dakako bili oni koji su najviše zastupljeni u samom programiranju, od same motivacije pa do sličnosti programa, dostupnosti u različitim okruženjima i onog najvažnijeg, stope ne odustajanja u učenju programiranja koja je jako važna karika za razvitak logičnog razmišljanja i pripreme na stvaranje budućih tehnologija.

Nakon prikupljenih svih podataka, napravljenih poglavlja i određenih alata smještenih u zasebne kategorije viđen je veliki broj korisnih informacija nego što se činilo da će se uočiti pri samom startu pisanja ovog rada.

Upravo ovakva analiza pruža uvid u dostupnost alata, njihove glavne karakteristike te zadaće i sam rad je bez obzira na početno znanje o programiranju dobar uvod za pregled osnovnih alata koji su dostupni široj publici i čitateljima koji svoje vrijeme i zanimanje žele posvetiti programiranju.

Ključne riječi: analiza, alati, programiranje, učenje, igre, podaci, motivacija

Abstract

During any type of comparative analysis, including tools intended for teaching programming, a vast amount of information is needed so that the used tools can be placed side by side with specific criteria. Web sites, forums and blogs were under the spotlight for a while as they can be used as guides to determine which tools are helpful and which ones are too complicated for some users or are inadequate. However, the majority of forums and blogs did not help in the making of the analysis, instead they were more like some sort of introduction to define what exactly are tools used for and how they work.

The vast majority of useful data used to make the analysis possible can be found in scientific papers, papers from various conferences, thesis from both bachelor's and master's degree and using specific professional terminology and references.

Criteria are the most important factors in programming. From the motivation to the similarities of the programs, accessibility in various surroundings and, most importantly self motivation to not give up from learning programming. Self motivation is the most important factor as it's the base of the development of logic thinking and the creation of future technologies.

After all the information has been gathered, after all the created chapters and specific tools that has been arranged in different categories, the amount of information that can be found is far greater than expected.

This type of analysis is an example of how many tools are available and their basic properties. Homeworks and training at any level or knowledge in programming are a good introduction for a general view of the basic tools that are available to the public and the readers that want to spend their free time programming.

Keywords: *analysis, tools, programming, learning, games, data, motivation*