Stvaranje edukativnih mobilnih aplikacija u osnovnoj školi

Dominković, Iva

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Teacher Education / Sveučilište u Zagrebu, Učiteljski fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:147:741967

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-16**

Repository / Repozitorij:

University of Zagreb Faculty of Teacher Education - Digital repository

SVEUČILIŠTE U ZAGREBU

UČITELJSKI FAKULTET

ODSJEK ZA UČITELJSKE STUDIJE

Iva Dominković

STVARANJE EDUKATIVNIH MOBILNIH APLIKACIJA U OSNOVNOJ ŠKOLI

Diplomski rad

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU

UČITELJSKI FAKULTET

ODSJEK ZA UČITELJSKE STUDIJE

Iva Dominković

STVARANJE EDUKATIVNIH MOBILNIH APLIKACIJA U OSNOVNOJ ŠKOLI

Diplomski rad

Mentor rada:

izv. prof. dr. sc. Predrag Oreški

Zagreb, 2023.

SAŽETAK

Edukativne se aplikacije, kao posljedica razvoja tehnologije, sve češće koriste u osnovnoškolskom obrazovanju. Proizvodnjom svih softvera, pa tako i mobilnih aplikacija bavi se znanstvena disciplina softversko inženjerstvo. Kroz ovaj će rad biti predstavljen *online* sustav za programiranje aplikacija MIT App Inventor. U osnovnoj se školi učenici susreću s programiranjem u vizualno-blokovskom programskom okruženju Scratch. MIT App Inventor sustav je kojim bi se učenici mogli koristiti ukoliko se snalaze u Scratchu, a učenicima bi bio pravi izazov stvoriti aplikaciju za operacijski sustav Android ili iOS koju bi mogli koristiti na vlastitim mobilnim uređajima. Za potrebe rada izrađene su dvije mobilne aplikacije. Mobilna aplikacija „Informatika 3“ namijenjena je za nastavu informatike, a cilj joj je kroz interaktivne kvizove utvrditi i uvježbati usvojene nastavne cjeline trećeg razreda osnovne škole predviđene informatičkim udžbenikom. Aplikacija „Kalendar prirode“ izrađena je u korelaciji s kurikulumom predmeta Priroda i društvo. Svrha aplikacije jest interaktivno vođenje kalendara prirode u jesen. Razvoj aplikacija, način uporabe i njihova svrha bit će opisane u ovome radu.

Ključne riječi: obrazovanje, edukativne mobilne aplikacije, softversko inženjerstvo, programiranje, MIT App Inventor

SUMMARY

Educational applications are increasingly used in elementary schools due to technological development. Development of all types of software, including mobile applications, is managed within the scientific discipline of software engineering. This thesis presents MIT App Inventor as an online system for programming mobile applications. In elementary school, students are introduced to programming through a visual block-based programming environment called Scratch. The MIT App Inventor system is designed for students who are familiar with programming in Scratch, enabling them a challenge to create their own mobile applications for Android and iOS operating systems that they can use on their own mobile devices. Two mobile applications were developed for the purposes of the thesis. The mobile application "Computer science 3" provides interactive quizzes for practicing third-grade Computer Science topics. Mobile application "Calendar of Nature" is aligned with the Science curriculum. The purpose of this application is to take notes about changes in nature during autumn. The thesis will describe the development of applications, their usage and purpose.

Key words: education, educational mobile applications, software engineering, programming, MIT App Inventor

Sadržaj

1. Uvod ... 1
2. Kurikulum nastavnog predmeta Informatika ... 3
2.1. Informacije i digitalna tehnologija ... 3
2.2. Računalno razmišljanje i programiranje ... 4
2.3. Digitalna pismenost i komunikacija ... 5
2.4. e-Društvo ... 6
3. Programiranje ... 7
4. Scratch ... 8
5. Code.org ... 9
6. Softversko inženjerstvo ... 10
6.1. Aplikacije ... 11
6.2. Procesi razvoja softvera i softverski modeli ... 12
6.2.1. Model vodopada ... 13
6.2.2. Model evolucijskog razvoja ... 14
6.2.3. Model inkrementalnog razvoja ... 15
6.2.4. Model formalnog razvoja ... 16
6.2.5. Model grafičkog razvoja ... 17
6.2.6. Model usmjeren na ponovnu upotrebu ... 17
6.3. Mobilne aplikacije u nastavi ... 18
7. MIT App Inventor ... 19
8. Izrada mobilnih aplikacija pomoću MIT App Inventora ... 19
9. Mobilna aplikacija „Kalendar prirode“ ... 21
9.1. Opis mobilne aplikacije „Kalendar prirode“ ... 22

9.1.1 Nizinski zavičaj 25
9.1.2 Brežuljkasti zavičaj 26
9.1.3 Gorski zavičaj 28
9.1.4 Primorski zavičaj 29
9.2. Ostvareni odgojno-obrazovni ishodi 31
9.3. Razvoj mobilne aplikacije „Kalendar prirode“ 32
10. Mobilna aplikacija „Informatika 3“ 41
10.1. Opis mobilne aplikacije „Informatika 3“ 41
10.2. Ostvareni odgojno-obrazovni ishodi 41
10.3. Razvoj mobilne aplikacije „Informatika 3“ 42
10.4. Mobilna aplikacija „Informatika 3“ u nastavi Informatike 59
11. Zaključak 61
12. Literatura 62
13. Prilozi i dodaci 64
14. Izjava o izvornosti rada 66

1. UVOD

U suvremenom svijetu brzog razvoja računalne znanosti učitelji imaju vrlo važnu ulogu odgoja i obrazovanja generacija odgovornih korisnika digitalnih tehnologija, ali i stvaratelja iste. Odgojno-obrazovni sustav Republike Hrvatske unutar osnovnoškolskog obrazovanja omogućava učenicima najranije dobi da stvaraju jednostavne programe i razvijaju se u svijetu informatike. Informatika je sadašnjost i budućnost, to nam pokazuje digitalizacija svega onoga što je prije nekoliko desetaka godina bilo nezamislivo. Razvojem tehnologije razvija se i digitalno društvo. Važnost Informatike, kao nastavnoga predmeta u osnovnim školama, neupitna je jer su djeca od svoje najranije dobi izložena digitalnim uređajima i velikom broju informacija. Upravo zato je vrlo važno ukazati učenicima na opasnosti i prijetnje s kojima se mogu susresti prilikom korištenja digitalnih uređaja i interneta, te ih naučiti kako se ispravno nositi s njima. Važno je ukazati i prednosti koje digitalna tehnologija donosi te otvoriti vrata mogućnostima koje ona donosi za njihov razvoj.

Programiranje se uvodi već u nižim razredima osnovne škole. Iako su učenici tada još usmjereni na konkretne primjere, važan je razvoj njihovog apstraktnog razmišljanja. Programiranje je dosta apstraktan proces o kojem će se kroz ovaj rad govoriti. Samu apstrakciju programiranja učenicima se približava zadatcima rješavanja problema koji razvijaju njihovo logičko razmišljanje, a zatim se započinje i proces programiranja i vizualno-blokovskom programskom okruženju kao što je Scratch. Kroz rad će biti detaljnije opisano programsko okruženje Scratch kao i njegova uporaba u nastavnom procesu predmeta Informatike. Programiranje se učenicima može približiti i pomoću neprofitne organizacije Code.org koja svake godine organizira kampanju *Hour of Code* kojom razbija stereotipe i širi raznolikost unutar svijeta informatike (code.org, 2023).

Mobilne aplikacije u nastavnom procesu više ne bi trebale biti stigmatizirane. Mogućnosti koje mobilne aplikacije pružaju korisnicima velike su, a zbog njihove jednostavnosti i dostupnosti, gotovo svaka učionica bi redovno mogla koristiti iste u sklopu nastave. Postoje brojne edukativne web i mobilne aplikacije koje se koriste u nastavi raznih predmeta. No, tu se ne treba zaustavljati. Kao što je već navedeno, obrazovanje nudi mogućnost stvaranja, a ne samo korištenja. Učenici mogu stvarati vlastite vrlo jednostavne aplikacije te ući u sam svijet programiranja na zabavan i koristan način.

Kroz ovaj rad opisat će se besplatno sučelje MIT App Inventor koje nudi mogućnost razvoja jednostavnih i složenijih aplikacija u vizualno-blokovskom programskom okruženju. MIT App Inventor odlično može poslužiti u nastavi Informatike za sve učenike koji su već upoznati s programiranjem u Scratchu.

Kako bi se aplikacije razvijale, potrebno je poznavati modele softverskoga procesa. Kroz rad će biti objašnjeno što je to softversko inženjerstvo te detaljniji opis sljedećih modela razvoja softvera: model vodopada, model evolucijskog razvoja, model inkrementalnog razvoja, model formalnog razvoja, model grafičkog razvoja te model usmjeren na ponovnu upotrebu.

Od navedenih modela posebno će se obratiti pažnja na model vodopada kojim su se razvile dvije mobilne aplikacije za potrebe ovog diplomskoga rada. Mobilna aplikacija „Kalendar prirode“ namijenjena je za učenike 2. i 3. razreda osnovne škole za nastavu Prirode i društva. Mobilna aplikacija „Informatika 3“ razvijena je za potrebe nastavnoga predmeta Informatike po udžbeniku „Moja domena 3“, a namijenjena je učenicima 3. razreda osnovne škole. U radu će se opisati razvoj mobilnih aplikacija, njihove mogućnosti korištenja te ishodi iz kurikuluma koji se njima ostvaruju.

2. KURIKULUM NASTAVNOG PREDMETA INFORMATIKA

Osnovno razumijevanje i poznavanje informatike u suvremenom dobu je prijeko potrebno za osobne, profesionalne i funkcionalne potrebe (Kafai, 2016). Kafai (2016) također navodi kako je u doba razvoja tehnologije važno da učenici razumiju sučelja, tehnologije i sustave kojima se svakodnevno koriste, a to se ujedno ostvaruje kroz Kurikulum nastavnoga predmeta Informatika.

Glavni dijelovi kurikuluma nastavnog predmeta Informatika su: Opis nastavnoga predmeta Informatika, Odgojno – obrazovni ciljevi učenja i poučavanja nastavnoga predmeta Informatika, Domene u organizaciji predmetnoga kurikuluma Informatike te Odgojno-obrazovni ishodi prema razredima i domenama (Ministarstvo znanosti i obrazovanja, 2018).

U opisu nastavnoga predmeta Informatika opisuje se važnost digitalne pismenosti te poznavanja temeljnih informatičkih koncepata u razdoblju velikog razvoja računalne znanosti. U ovom se dijelu kurikuluma definira pojam informatike u obrazovnom sustavu te koje se vještine i kompetencije razvijaju nastavnim predmetom (Ministarstvo znanosti i obrazovanja, 2018).

Ministarstvo znanosti i obrazovanja (2018) navodi odgojno-obrazovne ciljeve učenja i poučavanja nastavnog predmeta Informatika: informatičko opismenjavanje učenika, razvijanje digitalne mudrosti, kritičkog i računalnog razmišljanja, komuniciranje i odgovorno surađivanje u digitalnom okruženju te razumijevanje i primjenjivanje sigurnosnih preporuka.

Prethodno navedeni ciljevi realiziraju se unutar četiri domene: Informacije i digitalna tehnologija, Računalno razmišljanje i programiranje, Digitalna pismenost i komunikacija te e-Društvo. Svaka domena ima svoje odgojno-obrazovne ishode prema razredima koji se ostvaruju tijekom nastave u školskoj godini. Nastava Informatike održava se 70 sati godišnje u svim razredima osnovne škole (Ministarstvo znanosti i obrazovanja, 2018).

2.1. Informacije i digitalna tehnologija

Unutar domene Informacije i digitalna tehnologija razvija se razumijevanje obrazaca digitalnog prikazivanja podataka kao što su tekst, brojevi, zvuk, slike i videozapisi. Učenici unutar ove domene upoznaju temeljne koncepte rada računala i digitalnih uređaja, obrasce pohrane podataka te načine prijenosa digitalnih informacija (Ministarstvo znanosti i obrazovanja, 2018).

Unutar ove domene razvija se apstraktno mišljenje primjernom vizualizacije i simulacije prikaza pojednostavljenih modela rada računala.

Učenici će nakon prvoga razreda biti u mogućnosti prepoznati digitalnu tehnologiju, komunicirati s poznatim osobama u digitalnom okruženju, razlikovati oblike digitalnih sadržaja, digitalne uređaje te postupke kojima se stvaraju digitalni sadržaji (Ministarstvo znanosti i obrazovanja, 2018).

Nakon drugog razreda učenici će objašnjavati ulogu programa u uporabi računala te razlikovati oblike digitalnih sadržaja, uređaje i postupke stvaranja digitalnih sadržaja (Ministarstvo znanosti i obrazovanja, 2018).

Ishodi koji se ostvaruju nakon trećeg razreda osnovne škole su korištenje simbolima za prikazivanje podataka te objašnjavanje i analiziranje jednostavnih hardverskih/softverskih problema i poteškoća koji se mogu dogoditi prilikom njihove upotrebe. (Ministarstvo znanosti i obrazovanja, 2018).

Nakon četvrtog razreda učenici će unutar ove domene moći objasniti koncept računalne mreže, analizirati čimbenike koji razlikuju ljude od strojeva te se koristiti simbolima za prikazivanje podataka, analizirati postupak prikazivanja i vrednovati njegovu učinkovitost (Ministarstvo znanosti i obrazovanja, 2018).

2.2. Računalno razmišljanje i programiranje

Domena Računalno razmišljanje i programiranje ima veliku važnost jer učenici unutar nje uče stvarati računalne programe. Unutar ove domene razvija se sposobnost logičkog zaključivanja, modeliranja, apstrahiranja i rješavanja problema (Ministarstvo znanosti i obrazovanja, 2018). Ministarstvo znanosti i obrazovanja (2018) navodi kako programiranje kod učenika razvija samopouzdanje, preciznost i upornost te sposobnost komunikacije i suradnje.

Nakon prvog razreda učenici unutar ove domene rješavaju jednostavan logički zadatak te prate i prikazuju slijed koraka potrebnih za rješavanje nekog jednostavnog zadatka (Ministarstvo znanosti i obrazovanja, 2018).

Učenici će nakon drugog razreda u ovoj domeni ostvariti ishode analiziranja niza uputa koje izvode jednostavan zadatak te stvaranja niza uputa u kojemu upotrebljavaju ponavljanje (Ministarstvo znanosti i obrazovanja, 2018).

Ministarstvo znanosti i obrazovanja (2018) navodi kako će učenici nakon trećeg razreda biti u mogućnosti stvoriti program korištenjem vizualnoga okruženja u kojem se koristi slijedom koraka, ponavljanjem i odlukom te slagati podatke na koristan način.

Nakon četvrtog razreda učenik će stvarati program korištenjem vizualnog okruženja u kojem koristi slijed, ponavljanje, odluku i ulazne vrijednosti te rješavati složenije logičke zadatke s uporabom računala ili bez uporabe računala (Ministarstvo znanosti i obrazovanja, 2018).

2.3. Digitalna pismenost i komunikacija

Digitalna pismenost treba se razvijati od najranije dobi, a odnosi se na poznavanje mogućnosti aktualne tehnologije i računalnih programa, mogućnosti hardverskih i softverskih rješenja, vještinu suradnje i komunikacije u *online* okruženju (Ministarstvo znanosti i obrazovanja, 2018).

Nakon prve godine učenja predmeta Informatike unutar ove domene učenici će, uz podršku učitelja, koristiti predložene programe i digitalne obrazovne sadržaje te vrlo jednostavnim radnjama izrađivati jednostavne digitalne sadržaje (Ministarstvo znanosti i obrazovanja, 2018).

Nakon drugog razreda učenici prema savjetima učitelja odabiru uređaj i program za jednostavne školske zadatke, izrađuju digitalne radove kombiniranjem različitih oblika sadržaja te uz pomoć učitelja surađuju i komuniciraju s poznatim osobama u sigurnom digitalnom okruženju (Ministarstvo znanosti i obrazovanja, 2018).

Prema Ministarstvu znanosti i obrazovanja (2018), nakon trećeg razreda učenici će samostalno birati uređaj i program od predloženih te procijeniti načine njihove uporabe, izrađivati jednostavne digitalne radove, koristiti se sigurnim digitalnim okruženjem za komunikaciju u suradničkim aktivnostima te razlikovati uloge i aktivnosti koje zahtijeva suradničko online okruženje.

Nakon četvrtog razreda učenici odabiru prikladan program za zadani zadatak, smišljaju plan izrade digitalnog rada, izrađuju digitalni rad te ga vrednuju, zajednički planiraju i ostvaruju jednostavne ideje u digitalnom okruženju (Ministarstvo znanosti i obrazovanja, 2018).

2.4. e-Društvo

Ministarstvo znanosti i obrazovanja (2018) navodi kako je pristup digitalnom društvu pravo svakoga pojedinca, a nudi mu i mogućnosti za uporabu raznih e-usluga. Digitalna tehnologija napreduje i uvelike pomaže u pristupanju brojnim uslugama, kao i obavljanju obaveza. Kroz ovu domenu osposobljavaju se odgovorni sudionici e-društva koji učinkovito upotrebljavaju internet štiteći se od prijevara i elektroničkoga nasilja.

Učenici će se nakon prvog razreda pažljivo i odgovorno koristiti informacijskom i komunikacijskom opremom, štititi svoje osobne podatke te primjenjivati zdrave navike ponašanja tijekom rada na računalu (Ministarstvo znanosti i obrazovanja, 2018).

Ministarstvo znanosti i obrazovanja (2018) navodi sljedeće ishode koje će učenici ostvarivati nakon drugog razreda: prepoznaje i opisuje neke poslove koji se koriste informacijskom i komunikacijskom tehnologijom, koristi se e-uslugama u području odgoja i obrazovanja, analizira neke opasnosti koje mogu nastupiti pri uporabi računala i interneta te pravilno na njih reagira te se odgovorno ponaša pri korištenju sadržajima i uslugama na internetu radi zaštite osobnih podataka i digitalnoga ugleda.

Nakon trećeg razreda učenici će primjenjivati preporuke o preraspodjeli vremena u kojemu se koristi digitalnom tehnologijom za učenje, komunikaciju i zabavu te primjereno reagirati na svaku opasnost/neugodnost u digitalnom okruženju (Ministarstvo znanosti i obrazovanja, 2018).

Unutar ove domene učenici će, nakon četvrtog razreda, biti osposobljeni za istraživanje ograničenja uporabe računalne tehnologije, primjenjivanje uputa za očuvanjem zdravlja i sigurnosti pri radu s računalima te analiziranje širokog spektra poslova (Ministarstvo znanosti i obrazovanja, 2018).

3. PROGRAMIRANJE

Programiranje je kompleksna vještina koja se definira kao skup uputa datim računalu kako bi računalo obavljalo određene zadatke i aktivnosti (Abesadze i Nozadze, 2020). Programiranje zahtijeva različite faze kako bi se došlo do rješenja problema. Za početak potrebno je definirati i analizirati problem, zatim se planira rješenje koje se kodira te na samom kraju slijedi testiranje i evaluacija programa. Ovi se koraci u različitim literaturama opisuju na nešto drugačije načine, ali glavna ideja procesa stvaranja programa je ista (Abesadze i Nozadze, 2020). „Programiranje je računalom potpomognuto rješavanje problema, otklanjanje grešaka, razvijanje logičkog i računalnog razmišljanja, što podrazumijeva razvoj strategija za rješavanje problema koji se mogu odnositi i na neprogramerska područja.“ (Mladenović, 2019, 10).

Begel (2007) navodi kako je učenje programiranja u sklopu nastave Informatike odličan medij za razvoj novih ideja, izražavanje kreativnosti, učenje komunikacije i međusobne suradnje. Za poučavanje programiranja u nižim razredima osnovne škole važno je suočiti se s problematikom apstraktnih koncepata programiranja koji su učenicima te dobi teško shvatljivi. Upravo zbog toga početno programiranje, koje je samo po sebi apstraktno, treba učenicima prikazati na što konkretnijim primjerima. Programiranjem učenike polagano vodimo od konkretnoga prema apstraktnome, a samim time razvijamo apstraktno razmišljanje (Mladenović, 2019). Kako bi programiranje bilo uspješno, važan je i razvitak sposobnosti rješavanja problema. Računalno razmišljanje je skup procesa rješavanja problema, a jedan od zadataka nastave Informatike je i razvijanje računalnog razmišljanja. Mladenović (2019) navodi kako je za programiranje potrebna sposobnost rješavanja problema, ali isto tako programiranje uvježbava tu istu sposobnost. U nastavi je važno učenicima zadati zadatke primjerene njihovom stupnju kognitivnog i emocionalnog razvoja kako bi se izbjegle frustracije i odbojnost prema programiranju.

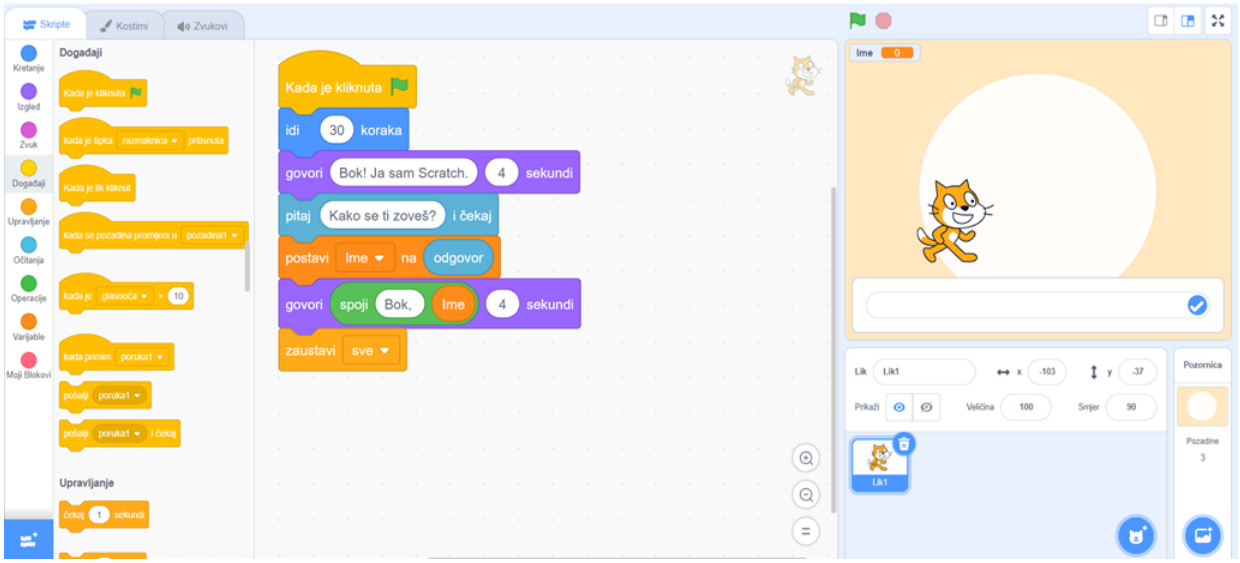
Za početno programiranje koriste se programski jezici jednostavne sintakse koji učenicima olakšavaju proces stvaranja programa. Mladenović (2019) dijeli programske jezike za početnike na: vizualno-tekstualne, vizualno-blokovske te proceduralno-tekstualne programske jezike.

Vizualno-tekstualni programski jezik smatra se primjerenim za početno programiranje učenika jer mu je značajka jednostavna tekstualna sintaksa s grafičkim rezultatom izvršavanja programa (Mladenović, 2019). Ovakvi programi najčešće se bave geometrijom što omogućava konkretno iskustvo programera. Primjer vizualno-tekstualnog programskog jezika je Logo.

Vizualno-blokovski programski jezici namijenjeni su za učenje i poučavanje programiranja djece mlađe dobi kao uvod u svijet programiranja. Ovakvi jezici imaju prikaz instrukcija u obliku blokova koji se mogu kombinirati poput slagalice, a izlaz je prikazan u obliku kretanja lika (Mladenović, 2019). Primjer vizualno-blokovskog programskog jezika je Scratch koji će biti detaljnije opisan sljedećem odlomku.

4. SCRATCH

Scratch je zajednica kodiranja jednostavnog vizualnog sučelja namijenjena djeci i mladima. Nastao je 2007. godine. Scratch je osmislila neprofitna organizacija Scratch Zaklada. Koristi se u školama za kreativno poučavanje i učenje, stvaranje digitalnih priča, igara i animacija. „Scratch promiče računalno razmišljanje i vještine rješavanja problema; kreativno poučavanje i učenje; samoizražavanje i suradnja; i jednakost u računalstvu.“ (scratch.mit.edu, 2023). Dostupan je na više od 70 jezika te se koristi za učenje programiranja diljem svijeta. Scratch je vizualno-blokovski programski jezik. Njegovo programsko okruženje prikazuje slika 1. U Scratchu se može programirati putem web preglednika, ali se mogu preuzeti i besplatna web aplikacija na računalo te mobilna aplikacija na pametne telefone ili tablete.



Slika 1. Vizualno-blokovsko programsko okruženje Scratch

Maloney (2010) navodi kako programsko okruženje Scratch odiše jednostavnošću zbog koje korisnici mogu vrlo lako mogu naučiti stvarati programe, a s druge strane ima dovoljnu dubinu i mogućnosti koje korisnike zadržavaju godinama.

Naredbe u Scratchu grupirane su u više kategorija. Kategorije naredbi su sljedeće: kretanje, izgled, zvuk, događaji, upravljanje, očitanja, operacije, varijable te moji blokovi. Blokove naredbi pretražuju se u kategorijama, a zatim se slažu u skripte poput slagalice. Blokovi su formirani na način da nije moguće spojiti dva bloka čija funkcionalnost nije moguća, što uvelike olakšava proces programiranja kod učenika.

Skripta je zapravo program koji se može pokrenuti određenom naredbom iz kategorije *Događaji*. U Scratchu postoji mogućnost izrade skripte za određenog lika, ali i za pozadinu. Likovi se mogu dodavati iz ponuđene baze likova unutar Scratcha, ali i uvođenjem novih likova s računala. Svaki se lik može urediti po želji. Isto pravilo vrijedi i za pozadinu koja se može mijenjati i uređivati po želji korisnika. Scratch nudi i opcije umetanja i snimanja zvuka koji se unutar programskoga jezika može uređivati. Prilikom programiranja u Scratchu učenici mogu u svakom trenutku testirati rad programa što im omogućava konkretan uvid u rad njihovog programa. Time se postiže brže otkrivanje mogućih pogrešaka i reagiranje na iste.

Scratch je odličan programski jezik za početke programiranja jer prati kognitivni razvoj djece i prepun je zornih predložaka koji omogućavaju lakše shvaćanje apstraktnoga procesa. Također, u Scratchu se mogu napraviti brojni zabavni i učenicima korisni programi, a sama misao učenika da su sami izradili program koji im služi njima je jako motivirajuća. Tomljenović (2018) navodi kako se događa prijelaz s igara za učenje na igre koje učenik aktivno izgrađuje, a upravo se to omogućuje programiranjem u Scratchu. Programiranje u Scratchu razvija računalno razmišljanje te je odlična priprema za programiranje u težim programskim jezicima (Wilson i Moffat, 2010).

5. CODE.ORG

Code.org je neprofitna organizacija čija misija je omogućiti svim učenicima diljem svijeta priliku za učenjem informatike. Code.org jednom godišnje organizira kampanju *Hour of Code* koja je uključila čak preko 15% svih učenika svijeta (code.org, 2023). Code.org dopire do učenika diljem svijeta i inspirira ih.

Ciljevi ove organizacije su: povećati raznolikost unutar informatičke znanosti, nadahnuti učenike i studente, postati dio redovne nastave, osposobiti i obučiti nove nastavnike informatike, promijeniti školski kurikulum te omogućiti dostupnost svojih usluga učenicima diljem svijeta. Code.org trenutno ima besplatne tečajeve dostupne na više od 67 jezika, koji se koriste u više od 180 zemalja (code.org, 2023).

Code.org pokrenuli su braća Hadi i Ali Partovi objavivši videozapis koji promovira računalnu znanost. U samo jednom danu taj je videozapis postao najgledaniji videozapis na YouTube-u što ih je potaknulo da počnu sakupljati volontere i stvore organizaciju. Od samih početaka pokrenuli su organizaciju uvjerenja kako svako dijete treba imati mogućnost kvalitetnog obrazovanja u svijetu informatike (core.org, 2023). Kako bi čitava neprofitna organizacija uspješno funkcionirala i širila svoje tečajeve i obuke djeci, mladima i odraslima diljem svijeta, ima podršku brojnih donora.

Hour of Code je globalni pokret koji pokreće Code.org, a cilj mu je kroz jednosatni uvod u kodiranje potaknuti razbijanje stereotipa i širenje raznolikosti unutar svijeta informatike. Njime se povećava pristup informatici te se prikazuje zabavna informatika koja je pristupačna bilo kojoj dobi. Važno je napomenuti i kako se ovdje ne radi o samo jednom satu kodiranja, već o brojnim događajima, tečajevima i aktivnostima koje se nude korisnicima ove organizacije (code.org, 2023).

Code.org može se koristiti u nastavi informatike kao besplatna mogućnost širenja znanja učenika, ali i nastavnika. Nastavnici mogu tijekom nastave upoznati učenike s ovom neprofitnom organizacijom i otvoriti im vrata raznih aktivnosti, tečajeva i događaja koje nudi Code.org.

6. SOFTVERSKO INŽENJERSTVO

Softversko inženjerstvo je znanstvena disciplina, ujedno i tehnička struka, koja se bavi razvojem i proizvodnjom softvera (Manger, 2016). Softver, odnosno softverski produkt je skup računalnih programa i pripadne dokumentacije razvijen za određenog korisnika ili tržište. Suvremeno tržište zahtijeva softvere koji su pouzdani, sigurni, učinkoviti, upotrebljivi te imaju mogućnost održavanja. Softversko inženjerstvo se bavi modelima, metodama i alatima za učinkovitu proizvodnju kvalitetnih softvera (Manger, 2016).

6.1. Aplikacije

Aplikacije su računalni programi koji omogućavaju korisnicima izvršavanje određenih aktivnosti. Postoje web aplikacije i mobilne aplikacije. Softversko inženjerstvo koristi metode, modele i alate za izradu aplikacije sukladno vrsti aplikacije koja se razvija (Sommerville, 2016). Postoje razne aplikacije koje zahtijevaju specijalizirane tehnike softverskog inženjeringa.

Sommerville (2016) navodi sljedeće vrste aplikacija: samostalne aplikacije, interaktivne aplikacije temeljene na transakcijama, ugrađeni sustavi upravljanja, sustavi serijske obrade, sustavi za zabavu, sustavi za modeliranje i simulaciju, sustavi za prikupljanje podataka te sustavi sustava. Važno je napomenuti kako granice među tim sustavima nisu jasno postavljene te se često kombiniraju različiti sustavi za izgradnju softvera s potrebnim karakteristikama.

Samostalne aplikacije su sustavi za osobna računala ili mobilne uređaje. Ova vrsta aplikacija ne mora nužno imati mrežnu povezanost kako bi ispunjavale sve potrebne zadatke. Primjeri samostalnih aplikacija su softveri za obradu fotografija, aplikacije za produktivnost, aplikacije za putovanja i slične (Sommerville, 2016).

Interaktivne aplikacije temeljene na transakcijama su aplikacije čije naredbe se izvršavaju s udaljenog računala, a kojima korisnici pristupaju s vlastitih digitalnih uređaja. Ova vrsta aplikacija vrlo je pogodna za *online* trgovine, poslovne sustave te sve one sustave koji imaju potrebu za pohranom mnoštva podataka koji brzo i lako ažuriraju svaku provedenu transakciju (Sommerville, 2016).

Ugrađeni sustavi upravljanja temelje se na kontroli i upravljanju hardverskim uređajima. Sommerville (2016) navodi kako su ugrađeni upravljački sustavi najvjerojatnije i brojčano najzastupljenija vrsta sustava. Primjer ugrađenog sustava upravljanja je softver koji upravlja procesom kuhanja u mikrovalnoj pećnici.

Sommerville (2016) sustave serijske obrade opisuje kao poslovne sustave razvijene za obradu velikih serija podataka. Primjeri sustava serijske obrade su sustavi za telefonske račune ili sustavi za isplatu plaća koji obrađuju velik broj pojedinačnih ulaza za stvaranje odgovarajućih izlaza podataka.

Sommerville (2016) navodi kako su sustavi za zabavu namijenjeni korisnicima za osobne potrebe i zabavu. Najčešći primjeri takvih sustava su raznovrsne igre od kojih se pojedine izvode na hardveru posebno namijenjenih igraćih konzola.

Sustavi za modeliranje i simulaciju razvijeni su od strane znanstvenika i inženjera kako bi prikazali modele pojedinih fizičkih procesa što je zahtijevalo brojne odvojene interaktivne komponente. Ovakvi sustavi se mogu izvoditi samo pomoću razvijenih sustava visokih performansi kako bi uspješno funkcionirali (Sommerville, 2016).

Sustavi za prikupljanje podataka služe za prikupljanje podataka iz okoline te slanje prikupljenih podataka drugim sustavima koji služe za obrađivanje istih. Softver komunicira sa senzorima koji se često nalaze unutar motora ili na udaljenoj lokaciji, a može uključivati i sustave koji prikupljaju i povezuju podatke unutar oblaka.

Sommerville (2016) navodi kako se sustavi sustava koriste u velikim organizacijama u kojima su potrebni sustavi sastavljeni od niza softverskih sustava. Jedan od primjera takvih sustava je generički softverski sustav ERP (*Enterprise Resource Planning*) za planiranje poslovnih resursa.

6.2.Procesi razvoja softvera i softverski modeli

„Softverski proces je skup aktivnosti i pripadnih rezultata čiji je cilj razvoj ili evolucija softvera.“ (Manger, 2016, 3). Manger (2016) kao osnovne aktivnosti, odnosno faze, unutar softverskog procesa navodi: utvrđivanje zahtjeva, oblikovanje, implementaciju, verifikaciju i validaciju te održavanje. Model za softverski proces prikazuje poželjan način odvijanja osnovnih aktivnosti, dok metoda razvoja softvera konkretizira odabrani model za softverski proces. Većina modela zasniva se na navedenim osnovnim aktivnostima. Utvrđivanje zahtjeva ujedno se naziva i specifikacijom, a tom se aktivnošću utvrđuje što softver treba raditi. Slijedi oblikovanje građe sustava te način rada njegovih dijelova. Implementacija, odnosno programiranje je realizacija oblikovanoga rješenja pomoću programskih jezika i alata. Nakon implementacije slijedi verifikacija i validacija, odnosno provjera samog rada softvera. Najčešći oblik verifikacije i validacije je testiranje. Nakon što softver uđe u upotrebu slijedi održavanje softvera koje se odnosi na popravke i nadogradnje u skladu s potrebama korisnika (Manger, 2016). Modeli se razlikuju po naglasku na određenim aktivnostima, a slijedi opis nekih modela za softverski proces.

6.2.1. Model vodopada

Model vodopada je softverski proces građen kao niz vremenski odvojenih aktivnosti, a slikovno je prikazan u obliku dijagrama (Manger, 2016). Sommerville (2016) navodi kako je model vodopada unaprijed isplanirani proces u kojem se planiraju i raspoređuju aktivnosti u procesu prije samog početka razvoja softvera. U ovom su modelu faze raspoređene na način da asociraju na vodu koja teče, odakle potječe i sam naziv modela (slika 2).

Treba naglasiti kako se mogućnost povratka u raniju fazu razvoja softvera koristi samo u iznimnim situacijama jer usporava i remeti tijek softverskoga procesa (Manger, 2016). Softverski proces prema modelu vodopada odvija se kroz sljedeće faze: specifikacija, oblikovanje, programiranje i testiranje modula, integracija modula i testiranje sustava te puštanje u rad i održavanje. Vidljivo je kako model vodopada prati osnovne faze softverskog procesa.

Specifikacija softvera, odnosno utvrđivanje zahtjeva utvrđuje se obzirom na potrebe korisnika sustava. Specifikacija je prva faza razvoja softvera te detaljno definira njegove ciljeve.

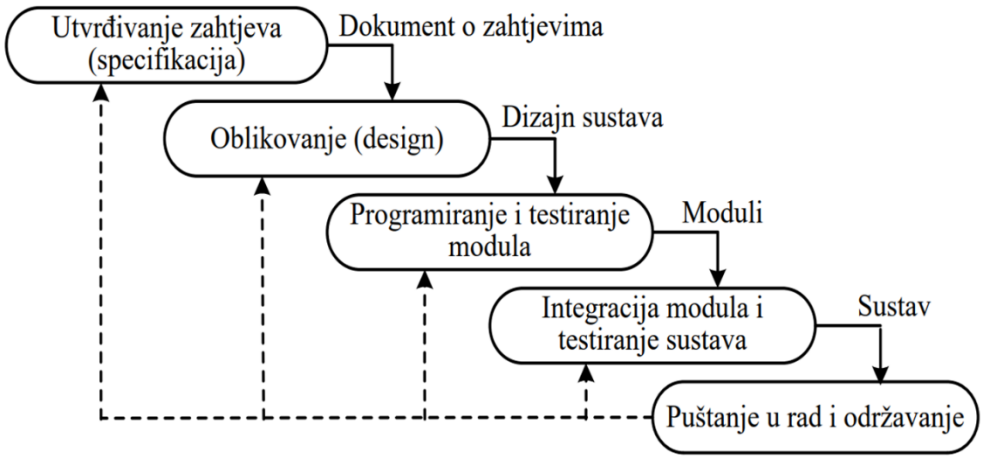
Druga faza razvoja softvera u modelu vodopada je oblikovanje, odnosno dizajn sustava u kojoj se uspostavlja arhitektura čitavog softvera. Oblikovanje softvera odnosi se na identificiranje i opisivanje temeljnih apstrakcija sustava i njihovih odnosa (Sommerville, 2016).

Sljedeća faza je programiranje i testiranje modula, a naziva se i implementacijom sustava. Unaprijed oblikovan softver se realizira kroz određene programe i programske jezike. Posebno se programiraju programske jedinice, odnosno moduli, a svaki se modul treba i testirati kako bi se provjerile njegove specifikacije.

Sommerville (2016) dalje opisuje četvrtu fazu procesa modela vodopada – integraciju modula i testiranje sustava. Cilj ove faze je integrirati pojedinačne programske jedinice te testirati cjelovit softverski sustav.

Peta, ujedno i posljednja faza je puštanje sustava u rad i njegovo održavanje. Nakon instalacije sustav se pušta u praktičnu upotrebu te započinje njegovo održavanje. Održavanje sustava odnosi se na ispravljanje pogrešaka te unapređenje softvera sukladno potrebama korisnika.

Proces izrade softvera modelom vodopada nerijetko je spor, a tome doprinose i teško razdvojive faze koje dovode do naknadnog otkrivanja pogrešaka. Unatoč tome, model vodopada je često korišten u praksi jer je pogodan za razvitak velikih sustava s jednostavnim i jasnim zahtjevima. Model vodopada omogućuje detaljno planiranje softverskog procesa i raspodjelu poslova na velik broj programera. Ovakav model izrade softvera omogućava lagan uvid u stanje u kojem se proces trenutno nalazi (Manger, 2016).



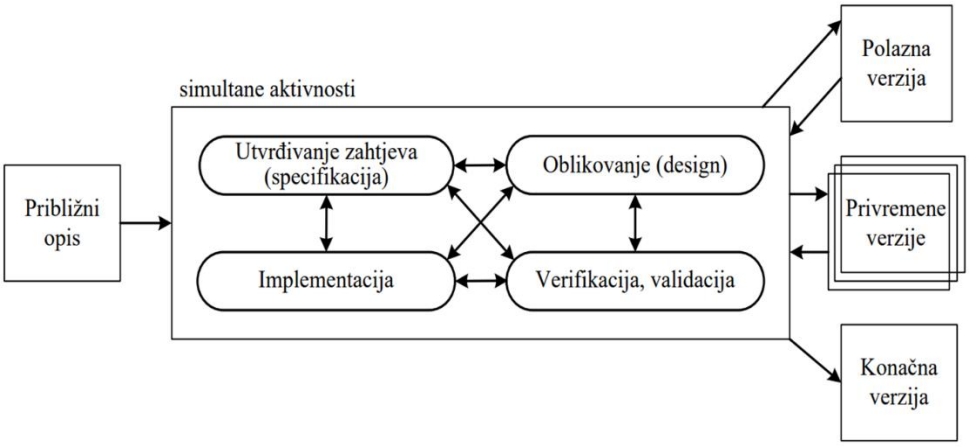
Slika 2. Softverski proces prema modelu vodopada (Manger, 2016, 6)

6.2.2. Model evolucijskog razvoja

Razvoj sustava po modelu evolucijskog razvoja (slika 3) odvija se na osnovi približnog opisa problema koje je polazište za nastanak prve verzije sustava. "Model evolucijskog razvoja (evolutionary development) nastao je kao protuteža modelu vodopada." (Manger, 2016, 7). Polazna se verzija pokazuje korisniku, a zatim se na temelju njegovih primjedbi i zahtjeva razvijaju poboljšane verzije sustava. Unutar svake iteracije simultano se obavljaju osnovne aktivnosti, a iteracije se odvijaju sve do konačne verzije sustava. Ovakav se model uspješno koristi za male sustave s nejasnim zahtjevima i kratkim životnim vijekom, kao i za razvoj web-sjedišta.

Model evolucijskog razvoja softvera ima prednost brzog izvođenja odgovora na zahtjeve korisnika čak i kada su zahtjevi nejasni. Specifikacija u modelu evolucijskog razvoja postepeno se razvija u skladu s korisnikovim razumijevanjem problema.

S druge strane, ovakav softverski proces nije prikladan za održavanje zbog najčešće loše strukturiranosti te nije moguće procijeniti kada će sustav biti gotov i u kojoj fazi izrade je. Također, Manger (2016) navodi kako su za razvoj sustava prema ovom modelu potrebni posebni alati i natprosječni softverski inženjeri.

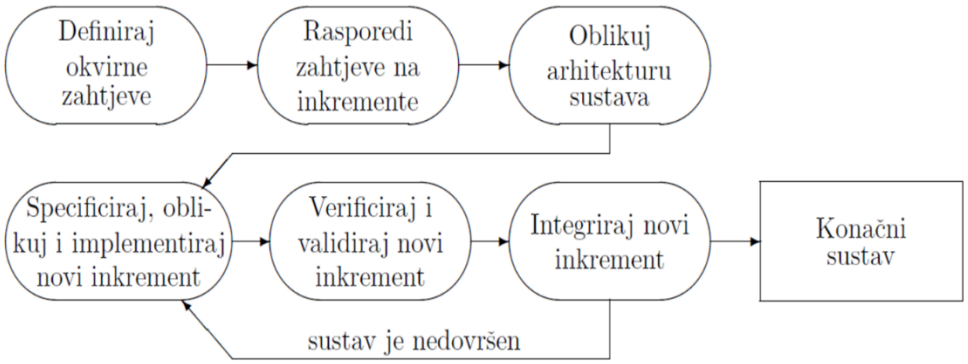


Slika 3. Softverski proces prema modelu evolucijskog razvoja (Manger, 2016, 7)

6.2.3. Model inkrementalnog razvoja

Model inkrementalnog razvoja može se shvatiti kao poboljšana verzija modela evolucijskog razvoja, odnosno kao hibrid modela vodopada i modela evolucije. Model inkrementalnog razvoja prikazan je na slici 4. Sustav se, kao i kod modela evolucijskog razvoja, razvija u nizu iteracija, no ne popravljaju se već realizirani dijelovi sustava, već se uvode novi dijelovi, takozvani inkrementi. Razvoj inkremenata unutar iteracija odvija se prema bilo kojem modelu (Manger, 2016).

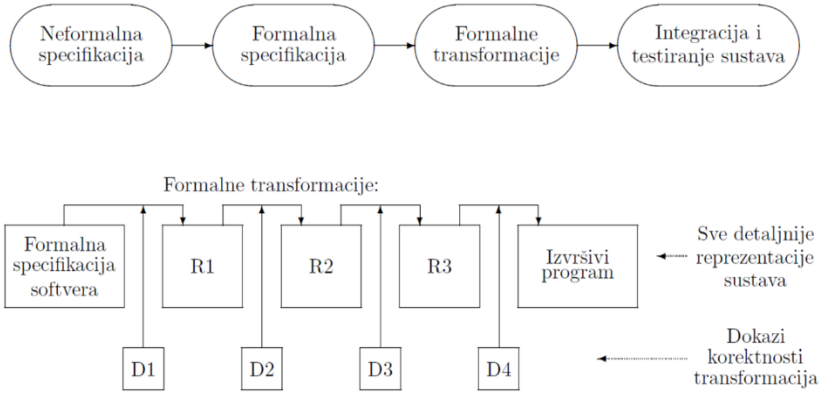
Model inkrementalnog razvoja transparentniji je od modela evolucijskog razvoja te je lakše prepoznati u kojoj se fazi razvoj sustava nalazi, no u ovakvom modelu je teže ispuniti sve korisničke zahtjeve koji nisu dovoljno razrađeni od samoga početka (Manger, 2016). Unatoč tome, ovaj se model koristi u praksi, a jedan od primjera sustava koji su se razvili prema ovom modelu su softverski paketi Microsoft Officea. Na slici 4 prikazan je model inkrementalnog razvoja.



Slika 4. Softverski proces prema modelu inkrementalnog razvoja (Manger, 2016, 8)

6.2.4. Model formalnog razvoja

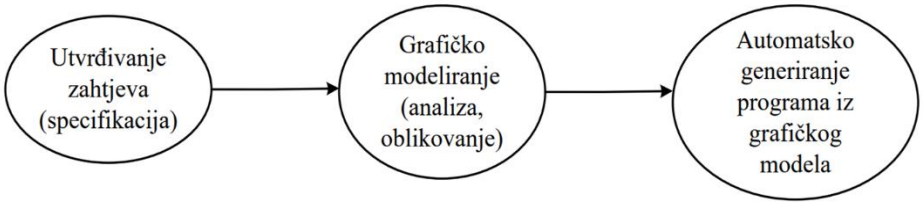
Model formalnog razvoja softvera, prikazan na slici 5, koristi se nedvosmislenim matematičkim notacijama kojima se precizno definiraju zahtjevi frontalnim načinom. Početna faza modela formalnog razvoja softvera je neformalna specifikacija, a slijede ju formalna specifikacija, formalne transformacije te integracija i testiranje sustava. Manger (2016) navodi kako se formalna specifikacija pretvara u program nizom koraka koji čuvaju korektnost. Ovakav razvoj softvera omogućuje automatiziranje faza razvoja nakon same izrade formalne specifikacije. Također, postoji matematički dokaz o točnoj implementaciji polazne specifikacije te je vrlo siguran i pouzdan način razvoja softvera, no vrlo malo se koristi u praksi jer zahtijeva veliki trud, znanje te trenutno nema dovoljno razvijenih programskih specifikacijskih jezika koji bi automatizirali transformacije (Manger, 2016).



Slika 5. Softverski proces prema modelu formalnog razvoja (Manger, 2016, 9)

6.2.5. Model grafičkog razvoja

Modelom grafičkoga razvoja prikazuje se građa i ponašanje sustava pomoću skupa dijagrama. Ovakav proces nastanka softvera trenutno se intenzivno istražuje, a za njega se koristi grafički jezik za modeliranje popu UML -a (Manger, 2016). Ovaj je model razvoja softvera sličan formalnom modelu, s time da bi trebao biti jednostavniji za korištenje. Trenutno se ne koristi, osim u jednostavnim primjerima, pošto trenutno nema dovoljno razvijenih alata koji su u mogućnosti generirati čitav program. Model grafičkoga razvoja prikazuje slika 6.

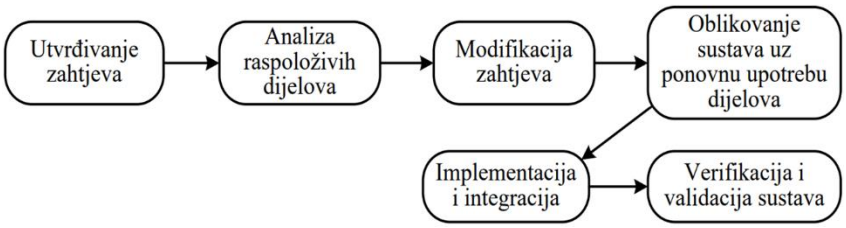


Slika 6. Softverski proces prema modelu grafičkoga razvoja (Manger, 2016, 10)

6.2.6. Model usmjeren na ponovnu upotrebu

Model usmjeren na ponovnu upotrebu koristi već postojeće upotrebljive softverske cjeline kao dijelove novoga softvera. Ideja ovakvog modela razvoja softvera je iskoristiti komponente ranije razvijenih sustava kako bi nastao novi sustav (Manger, 2016). Faze razvoja softvera ovim modelu su: utvrđivanje zahtjeva, analiza raspoloživih dijelova, modifikacija zahtjeva, oblikovanje sustava uz ponovnu upotrebu dijelova, implementacija i integracija te verifikacija i validacija sustava.

Prednost modela usmjerenog na ponovnu upotrebu, prikazanog na slici 7, je ušteda vremena jer se oslonac stavlja na provjerene i testirane dijelove sustava. Iako postoji mogućnost da sustav neće potpuno odgovarati zahtjevima korisnika zbog kompromisa u specifikaciji, očekuje se kako će se ovaj model u budućnosti jako puno koristiti (Manger, 2016).



Slika 7. Softverski proces prema modelu usmjerenom na ponovnu uporabu (Manger, 2016, 11)

6.3.Mobilne aplikacije u nastavi

Suvremeno obrazovanje zahtijeva i suvremene metode rada, kao i implementaciju informacijsko-komunikacijske tehnologije. Suvremena nastava zahtijeva razvijanje vještina učenika koje će ih osposobiti za korištenje informacijskih tehnologija u korist njihovog osobnog i profesionalnog razvoja (Begel, 2007). Informacijsko-komunikacijska tehnologija je dio svakodnevice današnjih generacija i upravo zato se u odgojno-obrazovnim ustanovama treba baviti i njome. Jedan od najvažnijih razloga tome je da će učenici puno svog slobodnog vremena provoditi uz digitalne uređaje i neophodno je ukazati im opasnosti i prijetnje s kojima se mogu susresti te kako se njima nositi. Još jedna prednost koju donosi korištenje tehnologije na raznim nastavnim predmetima je ta što će učenici uvidjeti kako im ona može biti vrlo korisna i kako se uz pravilno korištenje njome mogu kognitivno razvijati.

Podrug (2017) navodi kako mobilne aplikacije danas bilježe stalan rast preuzimanja radi svoje dostupnosti te jednostavnosti korisničkoga sučelja. Naglim porastom razvoja mobilnih uređaja, mobilne aplikacije postale su iznimno važan dio tržišta softvera. Mobilne se aplikacije mogu koristiti i u nastavi. Korištenjem mobilnih uređaja izbjegavamo problem nedovoljno opremljenih učionica i možemo na jednostavan i besplatan način pobuditi znatiželju učenika (Podrug, 2017).

Postoje razne web i mobilne aplikacije koje su edukativne naravi te se mogu koristiti u samoj nastavi, a učenici ih mogu koristiti i kod kuće. Korištenjem mobilnih aplikacija u nastavni povećava se motivacija učenika te njihova aktivnost. Odgojno-obrazovni proces nije samo suhoparna frontalna nastava i iziskuje brojne interaktivne sadržaje. Mobilne aplikacije omogućuju interaktivne sadržaje bez dodatnih troškova i priprema brojnih materijala. Učenicima je nastava uz korištenje mobilnih aplikacija zabavna i zanimljiva jer uče kroz igru te aktivno sudjeluju na satu. Učenje kroz igru učenike oslobađa straha od pogrešnog odgovora, a potiče ih se na istraživanje. Mobilne aplikacije mogu unaprijediti i upotpuniti nastavni proces, ali nikako ne bi trebale prevladavati. Odlične su za korištenje u motivaciji za nastavnim sadržajem, kao i za ponavljanje i uvježbavanje nastavne cjeline. Pojedini nastavni sadržaji mogu se i u potpunosti usvojiti isključivo pomoću mobilnih aplikacija. Također, aktivnosti koje nude edukativne mobilne aplikacije mogu poslužiti i kao zadatci za domaću zadaću.

7. MIT APP INVENTOR

MIT App Inventor je vizualno okruženje za programiranje koje omogućuje izradu funkcionalnih aplikacija namijenjenih za pametne telefone i tablete. Programsko okruženje MIT App Inventora nalikuje okruženju u Scratchu jer je temeljen na blokovima naredbi pomoću kojih se stvara program. Projekt koji pokreće MIT App Inventor nastoji približiti programiranje svim naraštajima, pogotovo mladima te omogućiti im da prijeđu na stvaranje tehnologije (appinventor.mit.edu, 2022). Profesor Hal Abelson predvodi MIT CSAIL tim osoblja i studenata koji zajedničkim snagama vode obrazovne aktivnosti MIT App Inventora, provode istraživanja o njihovim utjecajima te održavaju programsko okruženje. MIT App Inventor je besplatno okruženje kojim se služi preko 6 milijuna registriranih korisnika (appinventor.mit.edu, 2022).

Ova platforma ohrabruje učenike diljem svijeta da samostalno izrade svoje aplikacije koje podržavaju operativni sustavi Android i iOS (Patton, Tissenbaum i Harunani, 2019). Učenici koji znaju programirati pomoću Scratcha vrlo se lako mogu snaći i na ovoj besplatnoj platformi. MIT App Inventor može motivirati učenike za programiranje jer nudi mogućnost instaliranja mobilne aplikacije MIT AI2 Companion pomoću koje se mogu spojiti s računalom i testirati rad dosadašnjeg programa koji su stvorili. Takav način rada učenicima može biti vrlo zabavan, ali i koristan jer u svakom trenutku mogu otkriti potencijalne poteškoće u razvoju softvera.

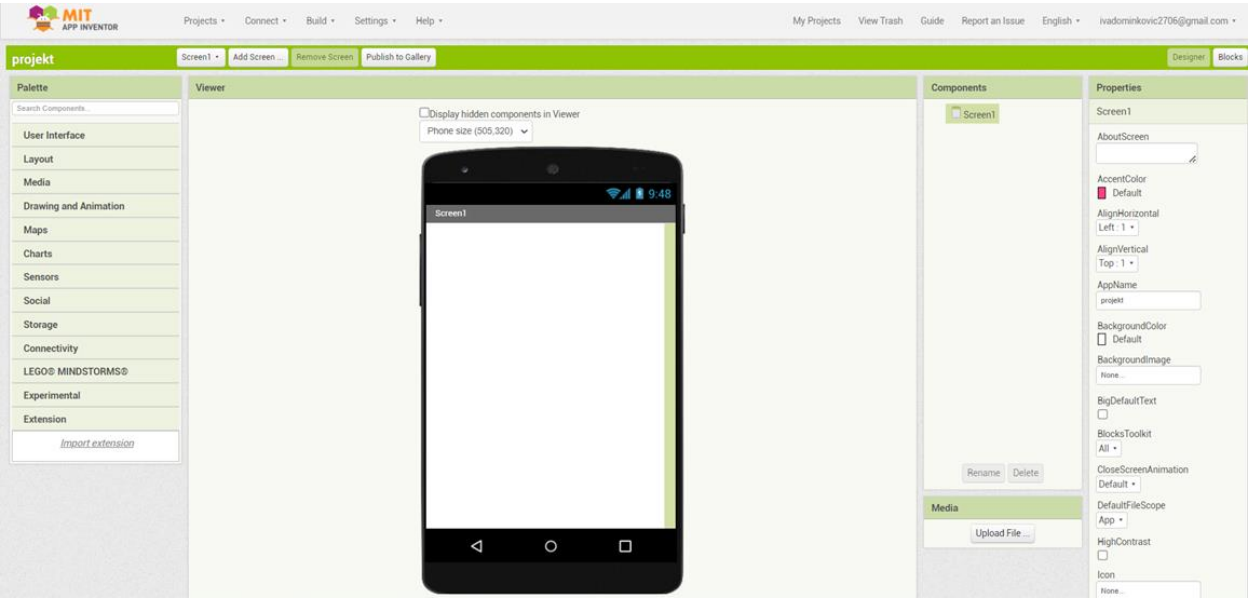
8. IZRADA MOBILNIH APLIKACIJA POMOĆU MIT APP INVENTORA

Korisničko sučelje MIT App Inventora sastoji se od dva glavna dijela: uređivača dizajna te uređivača blokova (Patton i suradnici, 2019).

Uređivač dizajna, odnosno dizajner (*Designer*) služi za vizualno raspoređivanje komponenata predloženih u takozvanoj paleti (*Palette*). Slika 8 prikazuje *Designer* u MIT App Inventoru.

Jedna aplikacija mora imati svoj početni zaslon, a može imati i više zaslona. Svaki zaslon treba sadržavati komponente od kojih se tvori njegov izgled, a zatim se za svaku komponentu dodaju naredbu kako bi cijeli program dobio funkciju. Svaki zaslon mora imati drugačije ime, a pomoću određenih naredbi aplikacija prilikom korištenja mijenja. Kompleksnije aplikacije mogu sadržavati više zaslona sastavljenih od brojnih komponenti koje dobivaju svoju funkciju pomoću vizualnih blokovskih naredbi. Svaki zaslon sadrži svoj program, odnosno niz blokova naredbi.

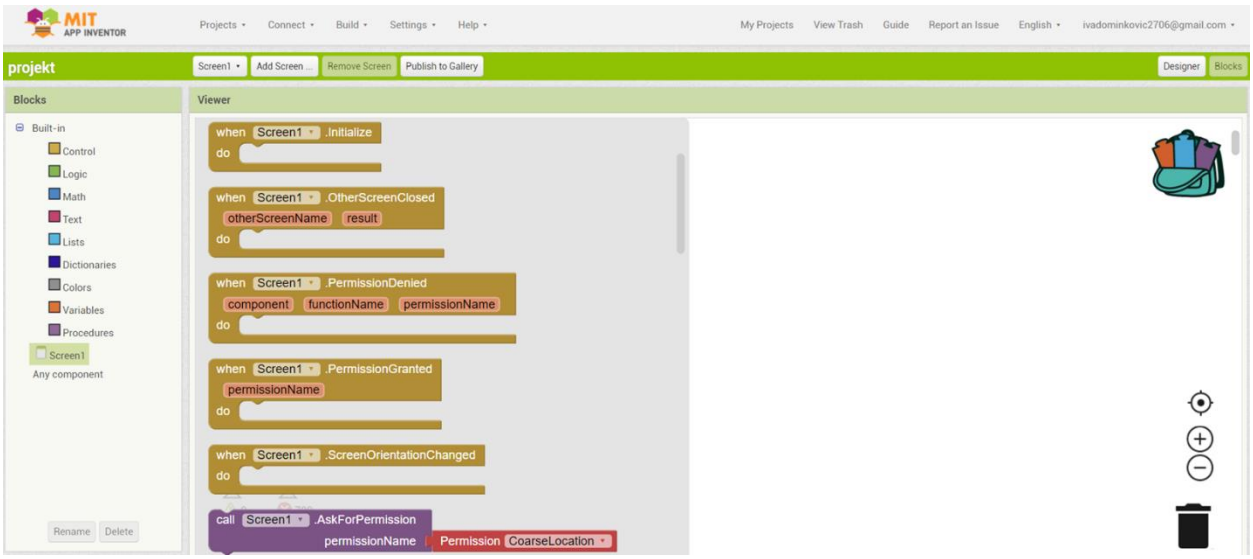
Korištenje komponenti u MIT App Inventoru omogućava apstrahiranje dijela složenosti ponašanja platforme (Patton i suradnici, 2019). Svaka komponenta ima svoj zaseban naziv kako bi se kasnije mogli dodavati blokovi naredbi za nju. Osim naziva, komponente imaju i svoja svojstva koja se mogu urediti prema želji korisnika. U uređivaču dizajna određuju se početna svojstva određene komponente kao što su pozadina, boja, veličina teksta, boja teksta, širina i duljina same komponente, njezina vidljivost i slično. Za dizajniranje zaslona koriste se vidljive komponente poput *Button*, *TextBox*, *Label* i slične, dok postoje i nevidljive komponente koje se na samom zaslonu ne vide, ali omogućuju pristup ugrađenim funkcijama uređaja (appinventor.mit.edu, 2022). Primjeri nevidljivih komponenti dostupnih u MIT App Inventor sučelju: *Accelerometer, Sound, OrientationSensor*. Komponente koje se koriste za dizajn same aplikacije raspoređene su prema kategorijama.



Slika 8. MIT App Inventor: *Designer*

Uređivač blokova (*Blocks*) sastoji se od predloženih blokova te platna na kojem se blokovi naredbi slažu u jednostavnije i kompleksnije programe. Prikaz blokova vidljiv je na slici 9. Korištenje blokova uklanja složenost temeljnog programskog jezika (Patton i suradnici, 2019). Blokovi su podijeljeni na dva glavna dijela: ugrađeni (*Built-in*) blokovi koji su uvijek isti te sastavni blokovi koji su predloženi za komponente s točno tog zaslona. Unutar ugrađenih blokova pronalaze se osnovne operacije dostupne u drugim programskih blokova (Patton i suradnici, 2019).

Ugrađeni blokovi raspoređeni su u više kategorija, vizualno razdvojenih po bojama, slično kao u programskom okruženju Scratch.



Slika 9. MIT App Inventor: *Blocks*

Blokovi se slažu u jednu vrstu slagalice, slično kao u okruženju Scratch te ukoliko dođe do potencijalnih problema izdaju se upozorenja na dnu platna koja se mogu prikazati. Također, platforma ne omogućava spajanje blokova koji funkcionalnu nisu spojivi, što od samoga početka programiranja sprječava pojedine poteškoće i probleme.

Kao što je već navedeno, MIT App Inventor može se koristiti u nastavi Informatike, ali i drugih nastavnih predmeta. Iako, su prvenstveno namijenjeni nastavnicima i učenicima srednjih škola (Patton i suradnici, 2019) mogu se iskoristiti i u osnovnoj školi. U razrednoj nastavi mogu se izrađivati vrlo jednostavni programi kao dijelovi jednostavnih aplikacija što može biti poticaj za istraživanje i daljnji razvoj u svijetu programiranja. Također, MIT App Inventor odlična je platforma i za nastavnike koji mogu samostalno razviti mobilnu aplikaciju prilagođenu dobi učenika i sadržajima nastavnoga predmeta za koji je namijenjena.

9. MOBILNA APLIKACIJA „KALENDAR PRIRODE“

Mobilna aplikacija „Kalendar prirode“ izrađena je za potrebe diplomskoga rada, a namijenjena je učenicima 3. razreda razredne nastave u sklopu predmeta Prirode i društva.

9.1. Opis mobilne aplikacije „Kalendar prirode“

Prilikom ulaska u aplikaciju „Kalendar prirode“ nudi se glavni izbornik od sastavnica: nizinski zavičaj, brežuljkasti zavičaj, gorski zavičaj te primorski zavičaj. Glavni izbornik aplikacije „Kalendar prirode“ nalazi se na slici 10. Učenici odabiru zavičaj u kojem žive kako bi unutar njega pratili svoj jednotjedni kalendar prirode. Izbornik sa zavičajima uveden je kako bi se pratilo načelo zavičajnosti. Učenici koji žive u različitim krajevima okruženi su različitim vremenskim prilikama, biljkama i životinjama, a shodno tome napravljeni su posebni obrasci za popunjavanje prikladni sva četiri zavičajna područja Republike Hrvatske. Tijekom izrade aplikacije pratila su se načela metodike predmeta Prirode i društva (De Zan, 2005).

Slika 10. Glavni izbornik aplikacije *Kalendar prirode*

Unutar svakog zavičaja nalazi se novi izbornik koji omogućava redni broj dana praćenja kalendara prirode. Postoji sedam mogućih odabira jer je kalendar prirode namijenjen za praćenje unutar sedam dana. Prilikom ulaska u svaki pojedinačni dan prikazuje se obrazac za ispunjavanje i praćenje opažanja učenika. Ova aplikacija namijenjena je za praćenje promjena u prirodi tijekom jeseni. Svaki obrazac unutar sebe ima dva vrlo važna gumba: gumb *Spremi* i gumb *Natrag*. Prilikom pritiska na gumb *Spremi* spremaju se sve dosada unesene promjene unutar obrasca što omogućava učenicima da im svaki obrazac bude spremljen onako kako su ga i ispunili. Komponentne se spremaju zasebno, stoga učenici ne moraju odjednom ispuniti čitav obrazac, već tijekom dana mogu ispunjavati komponentnu po komponentu uz uvjet da svaki puta spreme svoje promjene. Gumb *Natrag* omogućava odlazak na prethodni izbornik.

Prva opcija unutar svakog obrasca je postavljanje nadnevka (slika 11) za koji se prate promjene u prirodi. Učenici imaju mogućost pritiskom na sličicu kalendara dobiti pristup kalendaru te postaviti današnji nadnevak (slika 12).

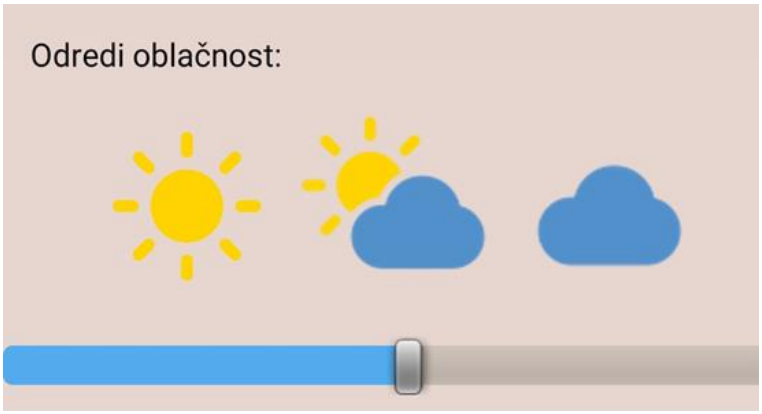
Slika 11. Obrazac za odabir nadnevka u aplikaciji *Kalendar prirode*

Slika 12. Opcija *DatePicker* u aplikaciji *Kalendar prirode*

Sljedeća opcija unutar svih obrazaca je odabir duljine dana, najniže i najviše temperature zraka pomoću opcije *Spinner*, prikazane na slici 13, koja omogućava odabir podatka iz ponuđene liste.

Slika 13. Primjer opcije *Spinner* u aplikaciji *Kalendar prirode*

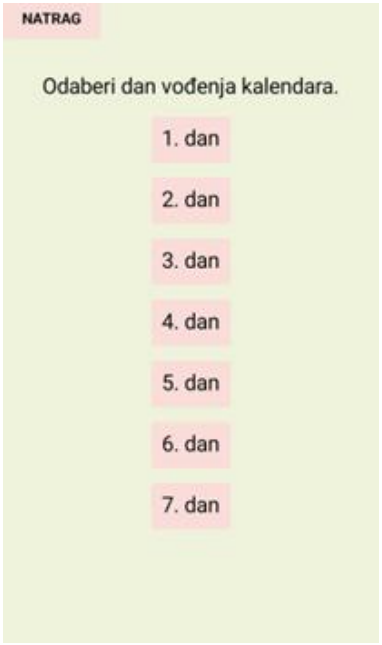
Također, unutar svakog obrasca učenici mogu odrediti oblačnost i jačinu padalina pomoću opcije *Slider* te označiti kućice ispod vremenskih prilika toga dana. Slika 14 prikazuje primjer opcije *Slider* za označavanje oblačnosti.



Slika 14. Primjer opcije *Slider* u aplikaciji *Kalendar prirode*

9.1.1 Nizinski zavičaj

Unutar izbornika Nizinski zavičaj nalazi se novi izbornik s mogućnosti odabira jednog od sedam dana. Izbornik za odabir dana unutar nizinskoga zavičaja prikazan je na slici 15.



Slika 15. Primjer izbornika za odabir dana u aplikaciji *Kalendar prirode*

Na samom vrhu stranice nalazi se gumb *Natrag* koji omogućava povratak u prethodni izbornik u bilo kojem trenutku rada.

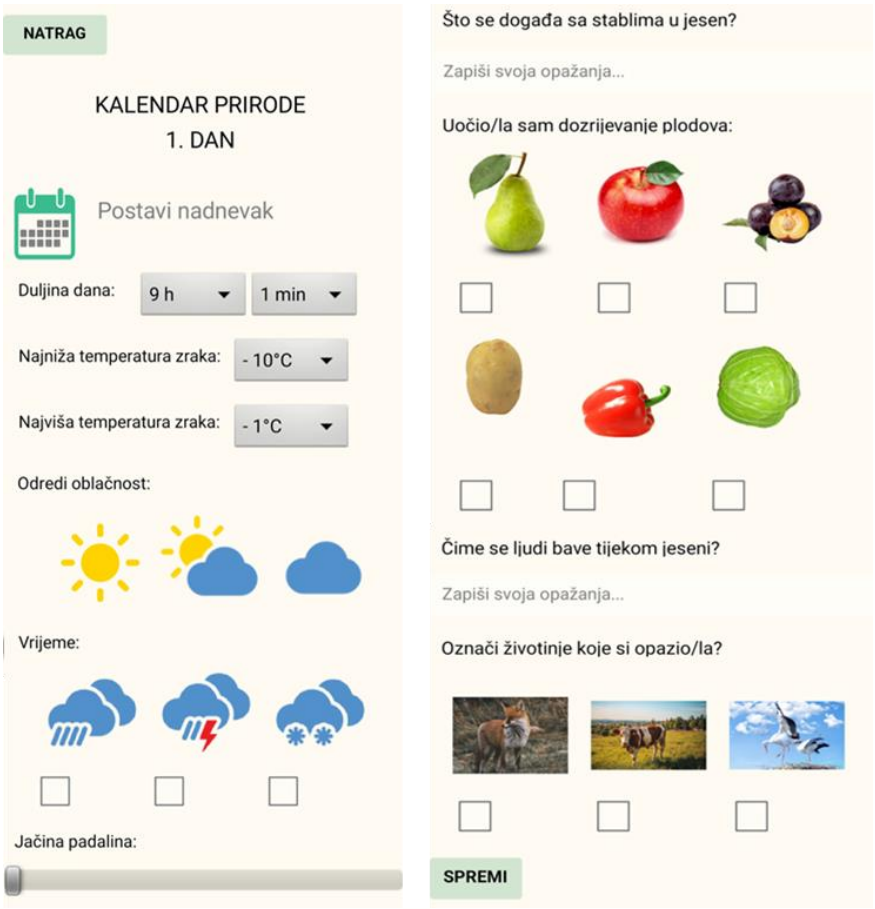
Nakon ponuđenih opcija koje su prisutne u sva četiri zavičaja, učenici koji žive u nizinskome zavičaju imaju ponuđeno mjesto za opis stabala u njihovom kraju. Učenici svakoga dana upisuju svoja zapažanja o tome što se događa sa stablima u jesen.

Slijedi opcija označavanja kućica ispod plodova čije dozrijevanje su učenici primijetili. Učenici imaju priliku označiti plodove specifične za njihov zavičaj: jabuku, krušku, šljivu, kupus, krumpir i papriku.

Zatim učenici imaju opciju unosa vlastitog teksta te opisuju čime se ljudi bave u jesen u njihovom zavičaju. Učenici trebaju povezati djelatnosti kojima se ljudi bave u nizinskome zavičaju s vremenskim prilikama, plodovima koji dozrijevaju te životinjama specifičnima za taj kraj.

Sljedeća opcija je ponovno označavanje kućica ispod životinja koje su učenici imali priliku opaziti u svome kraju: lisica, krava i roda.

Na samom kraju obrasca nalazi se gumb *Spremi* kojim učenici spremaju sve dosada izmijenjene i unesene podatke kako bi im prilikom ulaska u taj dan bio omogućen uvid u ono što su pratili. Slika 16 prikazuje obrazac za ispunjavanje jednoga dana ovoga zavičaja.



Slika 16. Prikaz obrasca jednoga dana nizinskoga zavičaja aplikacije *Kalendar prirode*

9.1.2 Brežuljkasti zavičaj

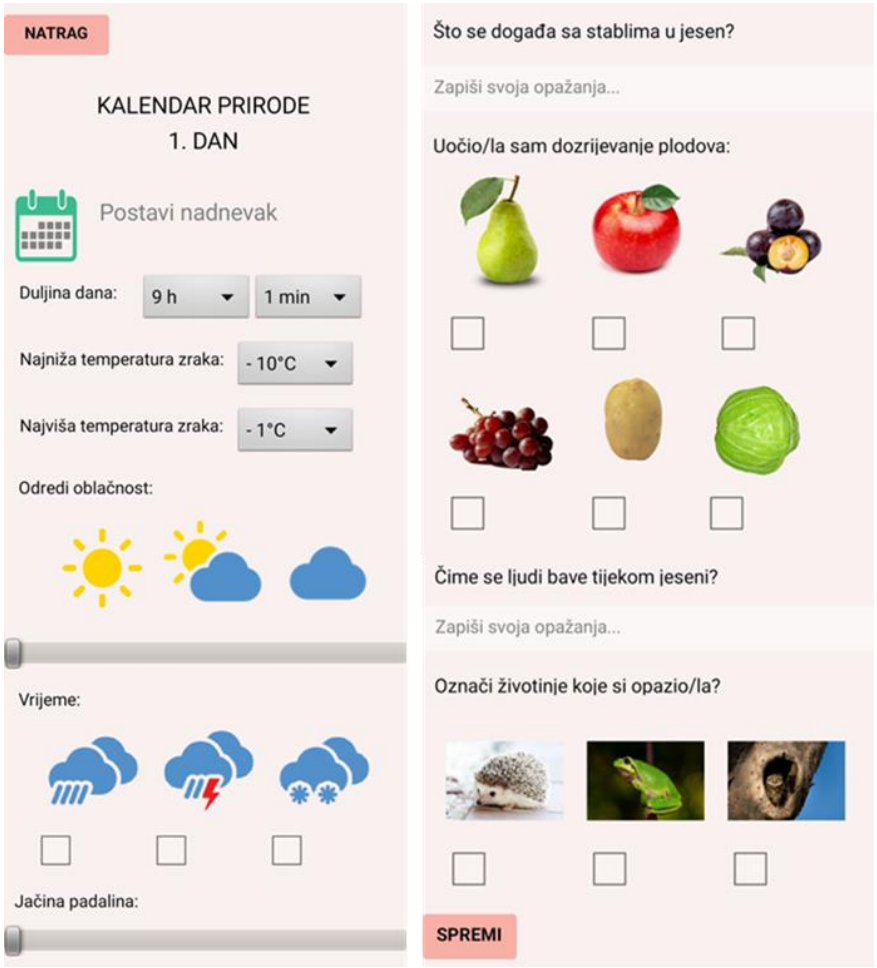
Kao i u prethodnom zavičaju na vrhu stranice nalazi se gumb *Natrag*, a na samome dnu stranice nalazi se gumb *Spremi*. Brežuljkasti zavičaj strukturiran je vrlo slično kao i nizinski.

Nakon opcija ponuđenih u svim obrascima, učenici brežuljkastoga zavičaja imaju mogućnost unosa vlastitog teksta kako bi napisali svoja opažanja o stablima.

Nakon te opcije učenici mogu označiti kućice ispod plodova čije dozrijevanje su primijetili u svome zavičaju. Učenici označuju pojedine od ponuđenih plodova: kruška, jabuka, šljiva, grožđe, krumpir i kupus.

Slijedi ponovan unos teksta u kojem učenici zapisuju svoja opažanja o djelatnostima ljudi u njihovom zavičaju te o tome čime se oni bave tijekom jeseni. Učenici povezuju djelatnosti ljudi brežuljkastoga zavičaja s vremenskim prilikama, dozrijevanjem plodova i životinjama koje su opazili u svome kraju.

Pred kraj samoga obrasca učenici imaju priliku označiti kućice ispod fotografija životinja koje su uočili u svome kraju: jež, žaba ili sova. Obrazac za ispunjavanje jednoga dana brežuljkastoga zavičaja prikazan je na slici 17.



Slika 17. Prikaz obrasca jednoga dana brežuljkastoga zavičaja aplikacije *Kalendar prirode*

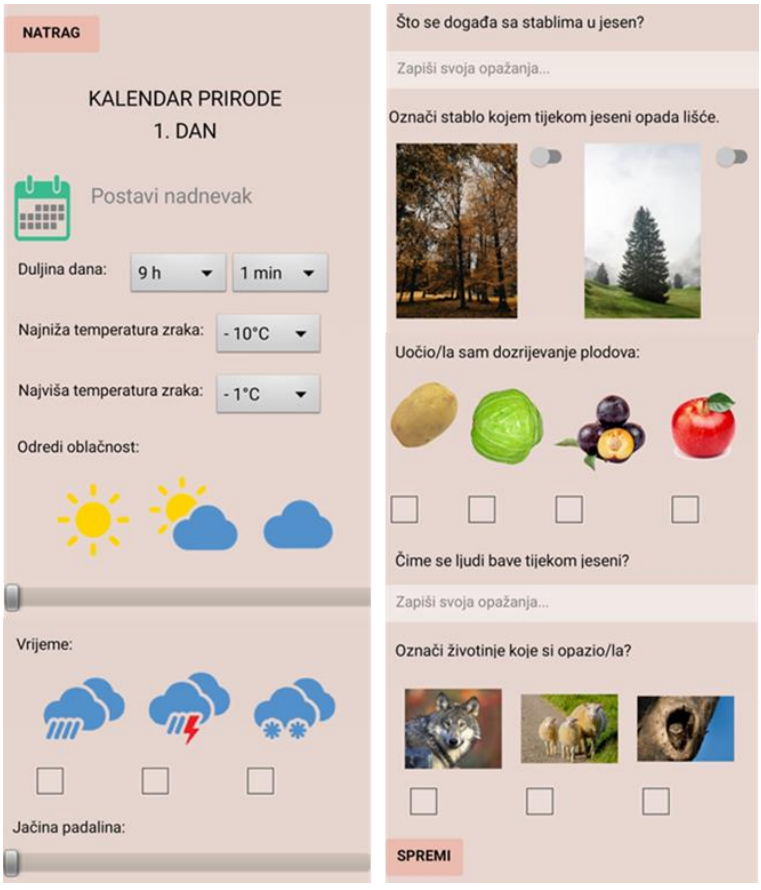
9.1.3 Gorski zavičaj

Obrazac gorskoga zavičaja (slika 18) započinje kao i svi obrasci do sada, a nakon zapisivanja opažanja o stablima, učenici imaju priliku označiti *Switch* pored fotografije stabla kojem tijekom jeseni opada lišće.

Nakon opažanja vezanih uz vremenske prilike i stabla, učenici označavaju plodove specifične za njihov zavičaj čije dozrijevanje su uočili. Ponuđeni su sljedeći plodovi: šljiva, krumpir, kupus i jabuka.

Slijedi obrazac s mogućnošću unosa vlastitoga teksta u kojem učenici zapisuju čime se ljudi njihovog zavičaja bave tijekom jeseni te povezuju to s djelatnostima toga zavičaja. Učenici povezuju djelatnosti ljudi s vremenskim prilikama, dozrijevanjem plodova i pojavom određenih životinja u tome kraju.

Na kraju obrasca učenici označavaju koje su životinje opazili od ponuđenih: vuk, ovca te sova.



Slika 18. Prikaz obrasca jednoga dana gorskoga zavičaja aplikacije *Kalendar prirode*

9.1.4 Primorski zavičaj

U aplikaciji pod primorskim zavičajem, uz temperaturu zraka, učenici mogu odrediti i prognozu temperature mora za taj dan.

U ovome zavičaju učenici umjesto opažanja vezana uz stabla, prate vjetar, odnosno njegovu brzinu i vrstu. Učenici tako mogu odrediti je li puhala bura ili jugo te prikazati jačinu vjetra opcijom *Slider*. Ovisno o uputama nastavnika, učenici mogu staviti svoju procjenu jačine vjetra ili je provjeriti na vremenskoj prognozi.

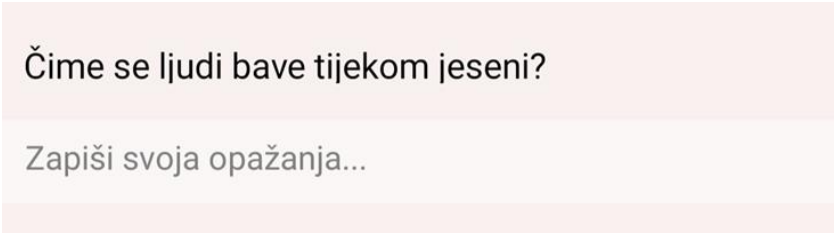
Slijedi označavanje plodova čije dozrijevanje su učenici primijetili. Za primorski zavičaj postavljene su fotografije sljedećih plodova: naranča, limun, mandarina, smokva, grožđe, masline, nar i nektarina.

Zatim učenici zapisuju svoja opažanja o tome čime se ljudi bave u njihovom zavičaju tijekom navedenog godišnjeg doba. Učenici, kao i su svim dosada opisanim zavičajima, povezuju ljudske djelatnosti toga zavičaja obzirom na vremenske prilike, dozrijevanje plodova i životinje specifične za taj zavičaj.

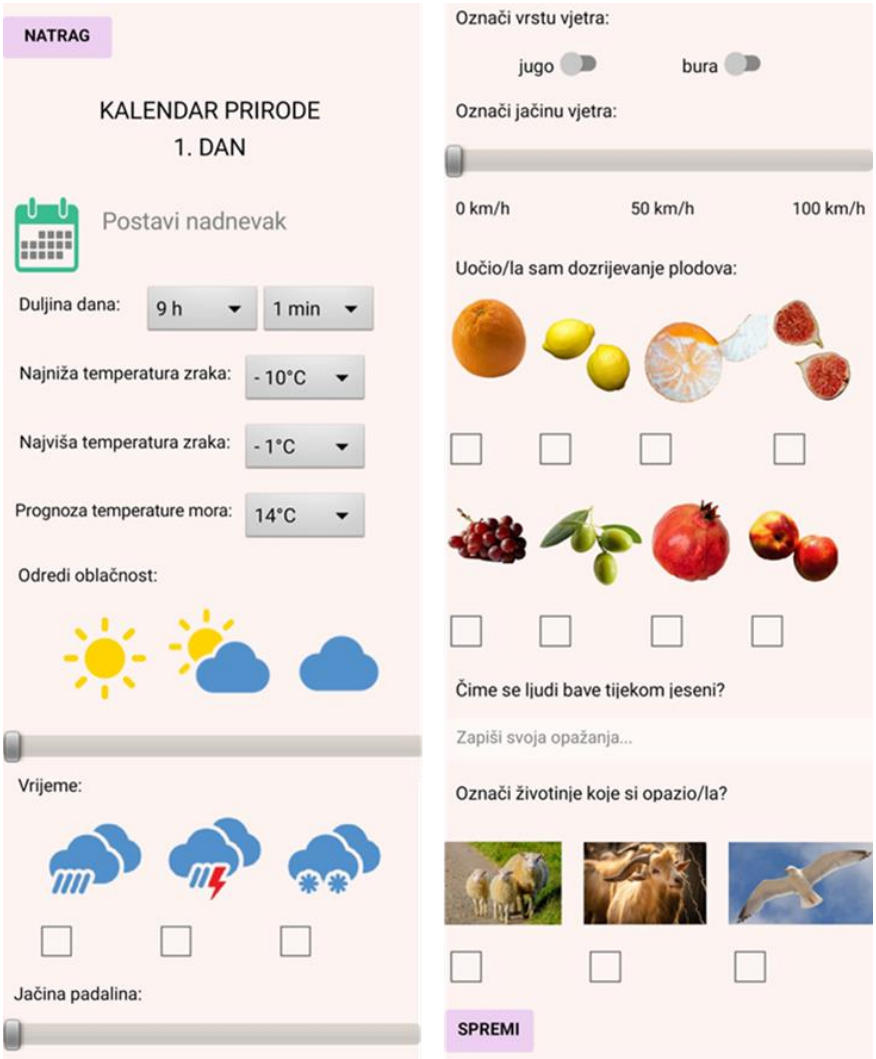
Pred kraj obrasca primorskoga zavičaja (slika 21) postoji mogućnost označavanja životinja koje su učenici opazili u svome kraju tijekom toga dana. Zatim spremaju svoj popunjeni obrazac kako bi mogli kasnije imati uvid u sve popraćene vrijednosti.

U svim se zavičajima za potrebe označavanja fotografija koristi opcija *CheckBox*, prikazana na slici 19 te mogućnost unosa zapažanja učenika pomoću *TextBoxa* prikazanog na slici 20.

Slika 19. Opcija *CheckBox* u aplikaciji *Kalendar prirode*



Slika 20. Opcija *TextBox* u aplikaciji *Kalendar prirode*



Slika 21. Prikaz obrasca jednoga dana primorskoga zavičaja aplikacije *Kalendar prirode*

9.2.Ostvareni odgojno-obrazovni ishodi

Aplikacija „Kalendar prirode“ može se koristiti u 2. ili 3. razredu osnovne škole u sklopu nastavnog predmeta Prirode i društva. Na samoj nastavi učitelji bi objasnili učenicima zadatak vođenja kalendara prirode, ali ih i upoznali s radom same aplikacije. Bilo bi idealno kada bi se jedan obrazac, barem većinski, ispunio tijekom same nastave kako bi se učenici lakše snalazili kod kuće. Nakon provedbe prikupljanja podataka na nastavi bi se podatci analizirali i izveli bi se zaključci.

Većina ishoda iz kurikuluma nastavnog predmeta Prirode i društva koji se ostvaruju su ishodi za 2. razred osnovne škole, no aplikacija može biti vrlo korisna i na početku 3. razreda kao ponavljanje i utvrđivanje brojnih gradiva iz 2. razreda. Posebno je pogodna za početak trećeg razreda jer se prate promjene godišnjega doba u kojem započinje školska godina – jeseni.

Ishodi iz kurikuluma za nastavni predmet Prirode i društva za osnovne škole u Republici Hrvatskoj (Ministarstvo znanosti i obrazovanja, 2019) koji se ostvaruju prilikom korištenja aplikacije „Kalendar prirode“ u drugom razredu osnovne škole:

PID OŠ A.2.1. Učenik uspoređuje organiziranost u prirodi i objašnjava važnost organiziranosti.

PID OŠ A.2.2. Učenik objašnjava organiziranost vremena i prikazuje vremenski slijed događaja.

PID OŠ A.2.3. Učenik uspoređuje organiziranost različitih zajednica i prostora dajući primjere iz neposrednoga okružja.

PID OŠ B.2.2. Učenik zaključuje o promjenama u prirodi koje se događaju tijekom godišnjih doba.

PID OŠ C.2.3. Učenik opisuje ulogu i utjecaj zajednice i okoliša na djelatnosti ljudi mjesta u kojemu živi te opisuje i navodi primjere važnosti i vrijednosti rada.

PID OŠ A.B.C.D. 2.1. Učenik uz usmjeravanje opisuje i predstavlja rezultate promatranja prirode, prirodnih ili društvenih pojava u neposrednome okružju i koristi se različitim izvorima informacija.

Ishodi iz kurikuluma za nastavni predmet Prirode i društva za osnovne škole u Republici Hrvatskoj (Ministarstvo znanosti i obrazovanja, 2019) koji se ostvaruju prilikom korištenja aplikacije „Kalendar prirode“ u trećem razredu osnovne škole:

PID OŠ A.3.1. Učenik zaključuje o organiziranosti prirode.

PID OŠ B.3.1. Učenik raspravlja o važnosti odgovornoga odnosa prema sebi, drugima i prirodi.

PID OŠ B.3.2. Učenik zaključuje o promjenama i odnosima u prirodi te međusobnoj ovisnosti živih bića i prostora na primjerima iz svoga okoliša.

PID OŠ A.B.C.D.3.1. Učenik uz usmjeravanje objašnjava rezultate vlastitih istraživanja prirode, prirodnih i/ili društvenih pojava i/ili različitih izvora informacija.

9.3. Razvoj mobilne aplikacije „Kalendar prirode“

Mobilna aplikacija „Kalendar prirode“ razvijena je prema modelu vodopada u besplatnom vizualnom programskom okruženju MIT App Inventoru. Aplikacija je izrađena 2023. godine za potrebe izrade diplomskoga rada.

Prva faza razvoja aplikacije „Kalendar prirode“ je specifikacija zahtjeva unutar koje su se odredili ciljevi sustava. Utvrdilo se kako će se razviti aplikacija za nastavni predmet Prirode i društva unutar koje će učenici voditi svoj jesenski kalendar prirode. Određeno je kako će aplikacija imati izbornik s četiri zavičajna područja Republike Hrvatske unutar kojih će se nuditi izbornik od sedam dana, odnosno sedam istih obrazaca za ispunjavanje opažanja učenika. Unaprijed je određeno koje će sastavnice učenici pratiti i bilježiti temeljem kurikuluma nastavnoga predmeta Priroda i društva i načela metodike. Za svako zavičajno područje usklađene su biljke i životinje koje pripadaju tom zavičaju kako bi se pratilo načelo zavičajnosti, sukladno dobi i mogućnostima učenika (Letina, predavanja). Utemeljeno je kako će kalendar prirode za svaki zavičaj imati unos nadnevka, najviše i najniže temperature zraka, vremenskih prilika te jačine padalina. Također, svaki će kalendar prirode unutar sebe sadržavati mogućnost označavanja plodova koji dozrijevaju u jesen u tom zavičaju, a plodovi se unutar obrazaca razlikuju ovisno o zavičajnom području. Utvrđeno je i kako će svako zavičajno područje imati mogućnost odabira životinja koje su uočili tijekom dana u svome području, a fotografije životinja usklađene su s podnebljem o kojem je riječ.

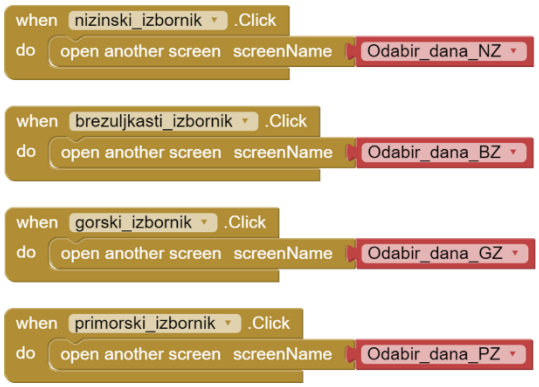
Unaprijed je određeno i kako će unutar primorskoga zavičaja učenici popratiti vrstu te jačinu vjetra, kao i temperaturu mora. Jedan od glavnih ciljeva aplikacije je mogućnost pohrane svih unesenih podataka i promijenjenih postavki kako bi učenici imali uvid u svoje praćenje prirodnih promjena tijekom jeseni.

Zatim, modelom vodopada, slijedi faza oblikovanja. MIT App Inventor, kao što je već opisano, ima dva glavna dijela sučelja – dio za oblikovanje i dio za programiranje. Faza oblikovanja odnosila se na sam dizajn aplikacije i njezinih stranica. Aplikacija se tvori od 33 stranice koje se izmjenjuju gumbu zadanim naredbama. Prva stranica *Screen* 1 je ujedno i početna stranica na kojoj se nalazi mogućnost odabira zavičaja pritiskom na jedan od četiri zadana gumba. Četiri gumba s početne stranice vode do četiri različite stranice koje služe za odabir rednog broja dana u kojem se prati kalendar prirode za svaki zasebni zavičaj.

Odabir dana vrši se pritiskom na jedan gumb od ponuđenih sedam. Svaki gumb odvodi do obrasca za točno jedan dan jednoga zavičaja. Za svaki zavičaj dizajniran je jedan obrazac za jedan dan, sukladan obilježjima zavičaja, a zatim se isti takav obrazac replicirao kako bi svaki zavičaj imao sedam jednakih obrazaca za sedam dana. U samom oblikovanju softvera koristile su se brojne mogućnosti MIT App Inventora kao što su: *Button, CheckBox, DatePicker, Image, Label, Slider, Spinner, Switch, TextBox, HorizontalArrangement, TableArrangement* te *TinyDB.*

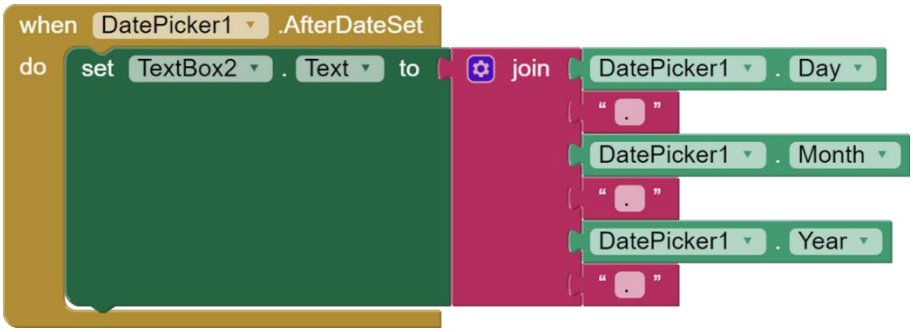
Tiny DB je nevidljiva komponenta MIT App Inventora koja pohranjuje podatke. Aplikacije stvorene pomoću MIT App Inventora svakim novim pokretanjem automatski se inicijaliziraju. Stoga je za pohranu podataka unesenih u aplikaciju „Kalendar prirode“ bilo nužno uvesti *TinyDB* kako bi se postavljene vrijednosti varijabli trajno pohranile. Kako bi to bilo moguće, potrebno je zadati tekstualni blok, odnosno naziv pod kojim će podatak biti pohranjen, a zatim ga pozvati određenom naredbom, koja je u slučaju aplikacije „Kalendar prirode“ pritisak na gumb *Spremi*.

Nakon oblikovanja zasebnih segmenata, odnosno modula, uslijedila je faza programiranja istih unutar dijela platforme za dodavanje i slaganje blokova naredbi. Na početnoj stranici (slika 22) svakom se gumbu dodala naredba se prilikom pritiska otvori nova stranica, odnosno ponuđeni zavičaj. Stranice za odabir dana, također imaju isprogramirane naredbe za otvaranjem obrazaca dana čiji je gumb pritisnut.



Slika 22. Prikaz programa početnog zaslona aplikacije *Kalendar prirode*

Za opciju unosa datuma postavljen je *DatePicker* (slika 23) koji omogućava odabir nadnevka iz kalendara. Kao ikona postavljena je sličica kalendara, a kako bi se datum zabilježio uveden je *TextBox* koji se ispunjava sukladno odabiru datuma. Pomoću baze podataka *TinyDB* navedeni datum ostaje spremljen pritiskom na gumb *Spremi*.



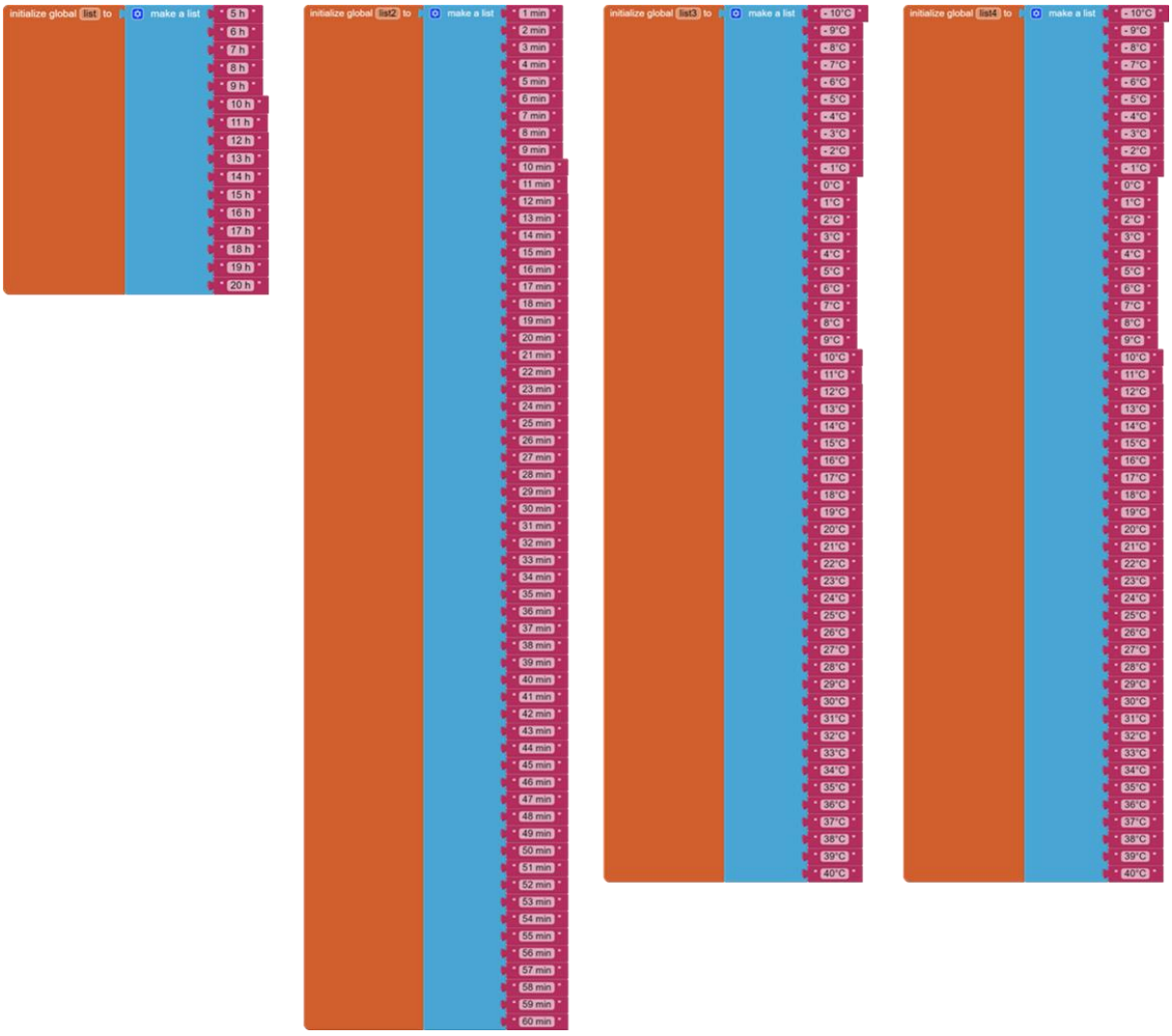
Slika 23. Prikaz programa za ispunjavanje nadnevka aplikacije *Kalendar prirode*

Kako bi se stvorile liste podataka za odabir temperature zraka, temperature mora i duljine dana koristila se mogućnost izrade globalne liste čiji su se elementi svrstavali u *Spinner*. Primjeri stvorenih globalnih lista prikazani su na slici 24. Svaki zasebni *Spinner* omogućava odabir ponuđenih elemenata obzirom na traženo. Uz pomoć baze podataka *TinyDB* odabrani element prilikom pritiska na gumb *Spremi* ostaje kao zadan.

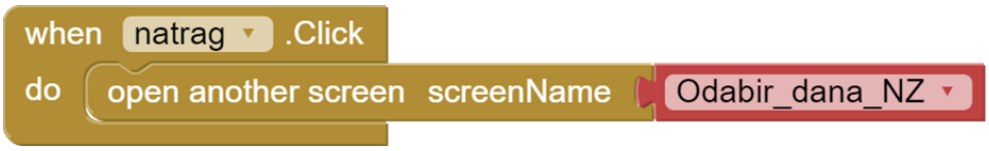
Jačina padalina, jačina vjetra te oblačnost određuje se pomoću opcije *Slider*. Svaki je *Slider* namješten na minimalnu početnu vrijednost, a pomoću baze podataka *TinyDB* se sprema na vrijednost koja je postavljena prstom korisnika.

Opcija *Switch* nalazi se u primorskom i gorskom zavičaju, a može se upaliti i ugasiti ovisno o potrebama korisnika. Tu opciju učenici također nakon promjene mogu pohraniti uz *TinyDB.*

CheckBox je postavljen ispod fotografija i slika koje učenici imaju mogućnost označiti. Koristi se za označavanja vremenskih prilika, plodova i životinja, a postavljena je mogućnost pohrane odabira korisnika uz *TinyDB.*

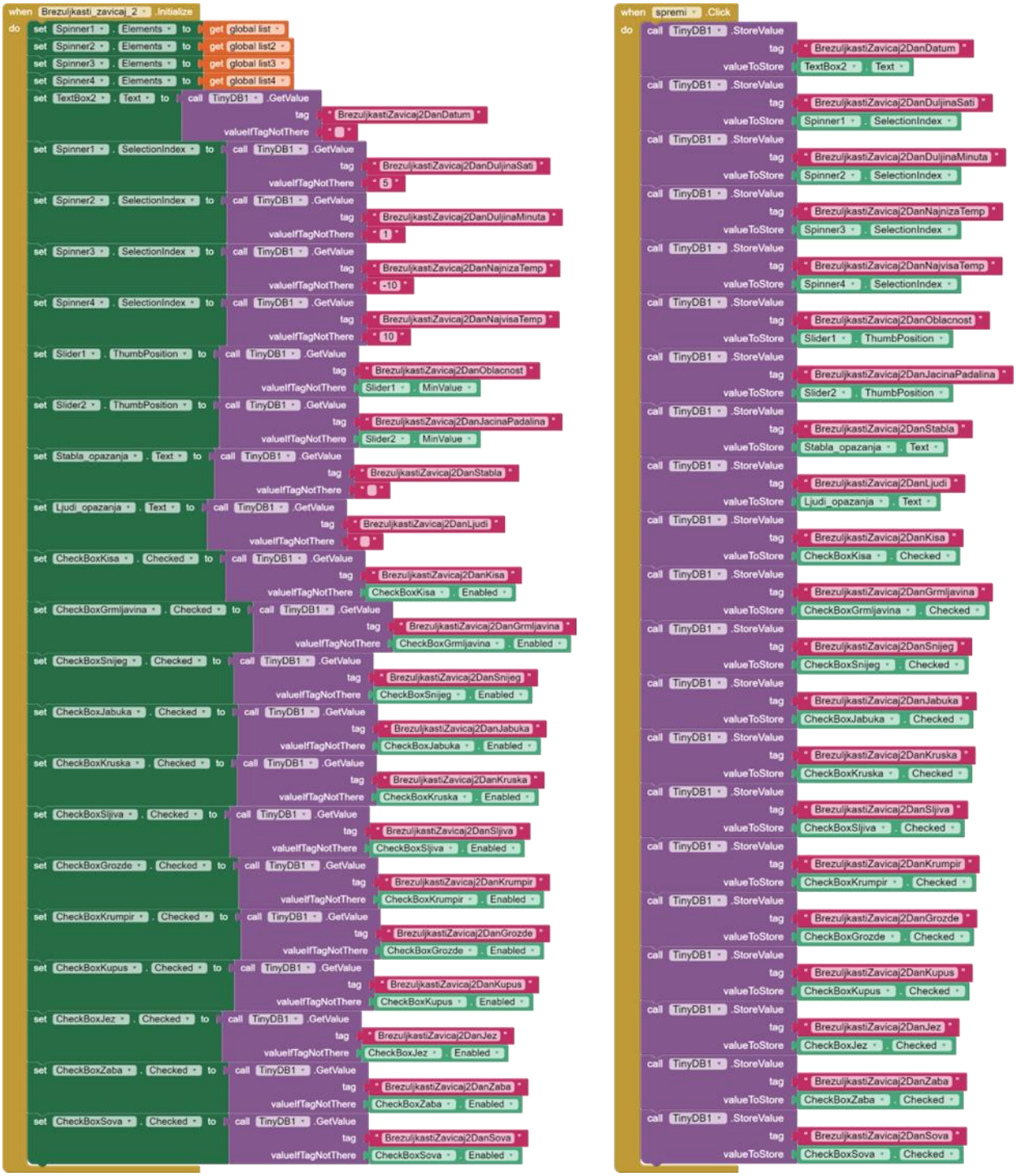


Slika 24. Globalne liste aplikacije *Kalendar prirode*

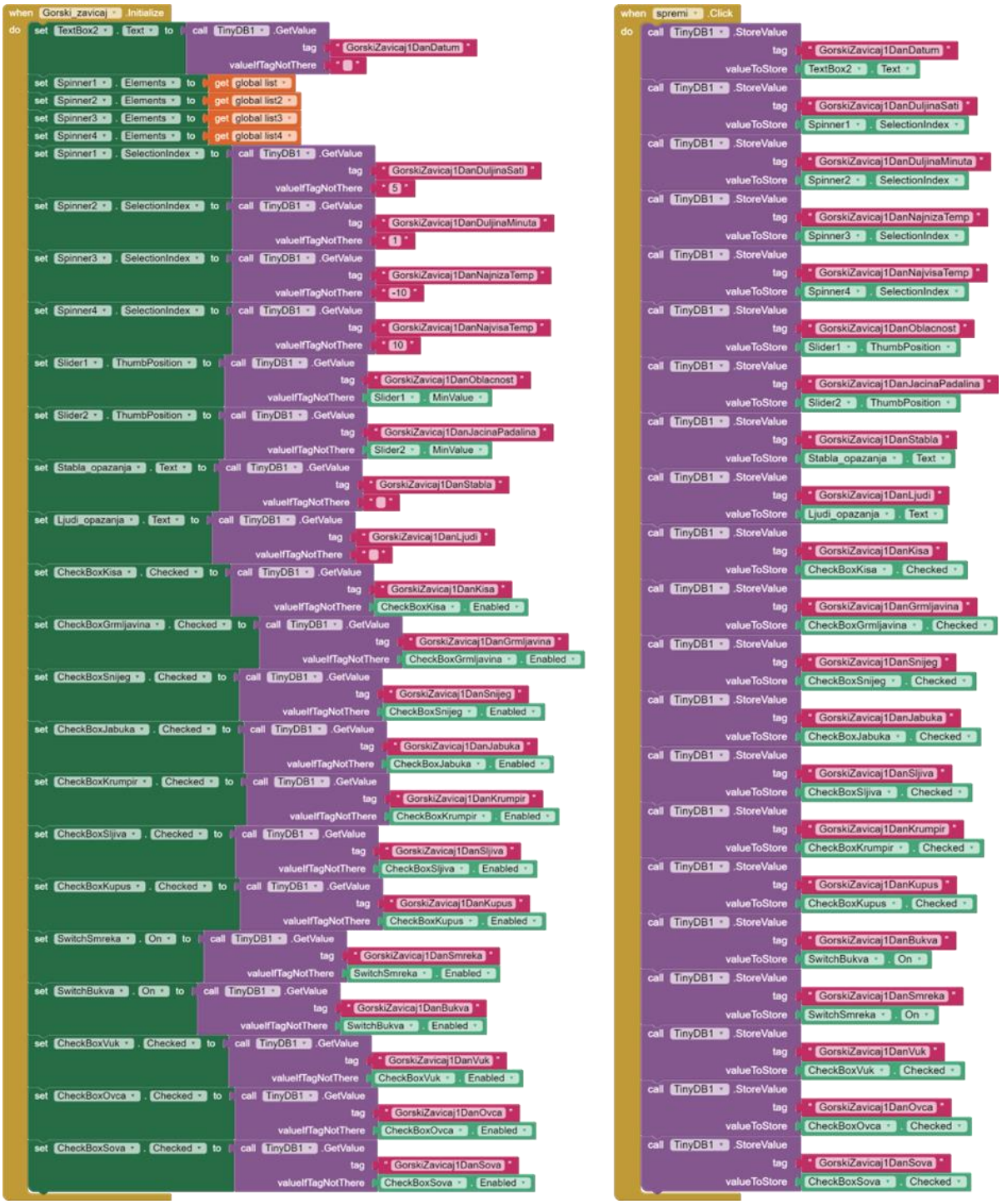


Slika 25. Primjer programa za povratak na izbornik *Odabir dana*

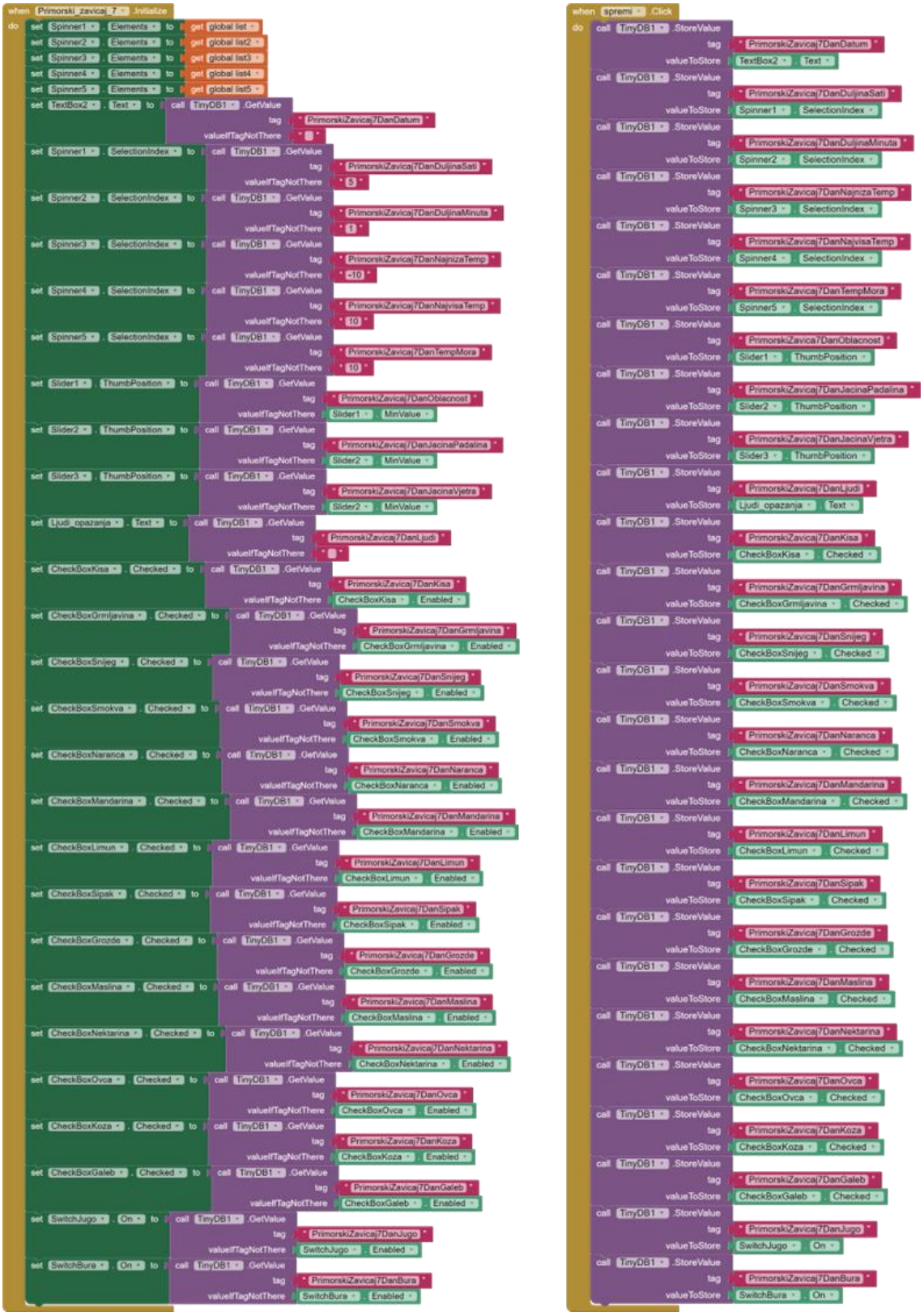
Slika 26. Primjer programa nizinskoga zavičaja za spremanje podataka u *TinyDB*



Slika 27. Primjer programa brežuljkastoga zavičaja za spremanje podataka u *TinyDB*



Slika 28. Primjer programa gorskoga zavičaja za spremanje podataka u *TinyDB*



Slika 29. Primjer programa gorskoga zavičaja za spremanje podataka u *TinyDB*

Nakon faze programiranja svakog modula, moduli su testirani pomoću aplikacije za mobilne uređaje MIT AI2 Companion koja omogućuje prikaz samog rada aplikacije dok ona još nije izgrađena i preuzeta na uređaj. Nakon oblikovanja i programiranja cijele aplikacije, rad aplikacije je u cijelosti testiran.

Nakon testiranja aplikacija se instalirala na mobilni uređaj putem QR koda koji MIT App Inventor nudi nakon same izgradnje aplikacije. Aplikacija je puštena u rad te započinje njezino održavanje.

10. MOBILNA APLIKACIJA „INFORMATIKA 3“

Mobilna aplikacija „Informatika 3“ razvijena je za potrebe diplomskoga rada, a namijenjena je učenicima 3. razreda za uvježbavanje i ponavljanje gradiva nastavnog predmeta Informatika.

10.1. Opis mobilne aplikacije „Informatika 3“

Mobilna aplikacija namijenjena je učenicima trećih razreda razredne nastave kako bi uvježbavali i ponovili gradivo cjelina iz udžbenika „Moja domena 3“. Unutar udžbenika „Moja domena 3“ gradivo je raspoređeno na sljedeće cjeline: Upoznajmo računalo, e-Svijet, Stvaranje digitalnog sadržaja, Rješavanje problema, Programiranje te Multimedija. Učenici prilikom ulaska u aplikaciju odabiru cjelinu koju žele ponoviti. Unutar svake cjeline nalazi se interaktivni kviz koji učenici mogu riješiti kako bi ponovili gradivo te cjeline. Aplikacija može služiti kao dio nastavnog procesa u školi ili kao zadatak za domaću zadaću prije provjere znanja. Prilikom izrade aplikacije pratila su se metodička pravila predmeta Informatika usvojena na predavanjima kolegija Metodika Informatike 1 i Metodika Informatike 2 (Dumančić).

10.2. Ostvareni odgojno-obrazovni ishodi

Prilikom specifikacije zahtjeva aplikacije, uzeli su se u obzir svi ishodi navedeni u kurikulumu nastavnog predmeta Informatike za učenike trećeg razreda osnovne škole. Ishodi su već detaljno opisani u odlomku Kurikulum nastavnog predmeta Informatika. Odgojno-obrazovni ishodi iz kurikuluma nastavnog predmeta Informatika ostvaruju se kroz cjeline udžbenika „Moja domena 3“ koji je koristio kao predložak za osmišljene zadatke koji se nalaze u kvizovima aplikacije.

10.3. Razvoj mobilne aplikacije „Informatika 3“

Mobilna aplikacija „Informatika 3“ razvila se prema modelu vodopada prateći metodička načela i ishode kurikuluma nastavnoga predmeta Informatike. Aplikacija je razvijena 2023. godine za potrebe diplomskoga rada, a namijenjena je za korištenje u nastavi Informatike.

Prvo se specificirao zahtjev aplikacije, odnosno odredio se njezina svrha i namjena. Cilj aplikacije je pružanje mogućnosti rješavanja kvizova za nastavne cjeline predviđene udžbenikom „Moja domena 3“. Nakon što su navedene cjeline utvrđene, analizom udžbenika odredile su se glavne značajke svake cjeline koje bi se trebale ispitati te su formirana pitanja za kviz svake cjeline. Određena je i mogućnost odgovora na pitanja te ponuđeni odgovori. Kada su sva pitanja i materijali bili spremni započelo je oblikovanje same aplikacije.

Za cjelinu „Upoznajmo računalo“ (slika 31) predviđeno je ispitati jesu li učenici usvojili koje su zdrave navike prilikom korištenja računala te koji je dnevni limit za provođenje uz digitalne uređaje. Također se provjerava razumiju li učenici razliku između stvarnoga i virtualnog svijeta, što su to simboli te kako reagirati na potencijalne poteškoće s kojima se možemo susresti prilikom korištenja digitalne tehnologije.

Iz cjeline „e-Svijet“ (slika 32) učenici utvrđuju kako se nositi s opasnostima i prijetnjama s kojima se mogu susresti na internetu. Kroz kviz ponavljaju što je to *cyberbullyng* te kako na njega reagirati te navode dobre i loše strane digitalnoga svijeta.

„Stvaranje digitalnog sadržaja“ je sljedeća cjelina unutar čijeg kviza (slika 33) učenici ponavljaju kako izgledaju ikone pojedinih programa i stavki na računalu. Prepoznaju ikone programa Paint, PowerPoint i WordPad te ikonu Google Chrome-a i Fotografija. Učenici razumiju funkciju mape te prepoznaju njezinu ikonu. Kroz ovu cjelinu učenici ponavljaju kako mogu kopirati i zalijepiti tekst, kako promijeniti font, urediti i poravnati tekst.

Unutar cjeline „Rješavanje problema“, čiji kviz je prikazan na slici 34, učenici imaju priliku razvijati logičko razmišljanje problemskim zadacima poput zagonetki, rebusa, nadopunjavanja niza, izbacivanja uljeza i sličnim.

Cjelina „Programiranje“ usredotočena je na ponavljanje opcija i mogućnosti u vizualnom programskom okruženju Scratch koje su učenici naučili. Odlična je kao kratko ponavljanje prije samoga rada u okruženju Scratch. Na dnu samoga kviza nalazi se i poveznica za vizualno programsko okruženje Scratch kako bi se nakon ponavljanja mogao pokrenuti praktični rad (slika 35).

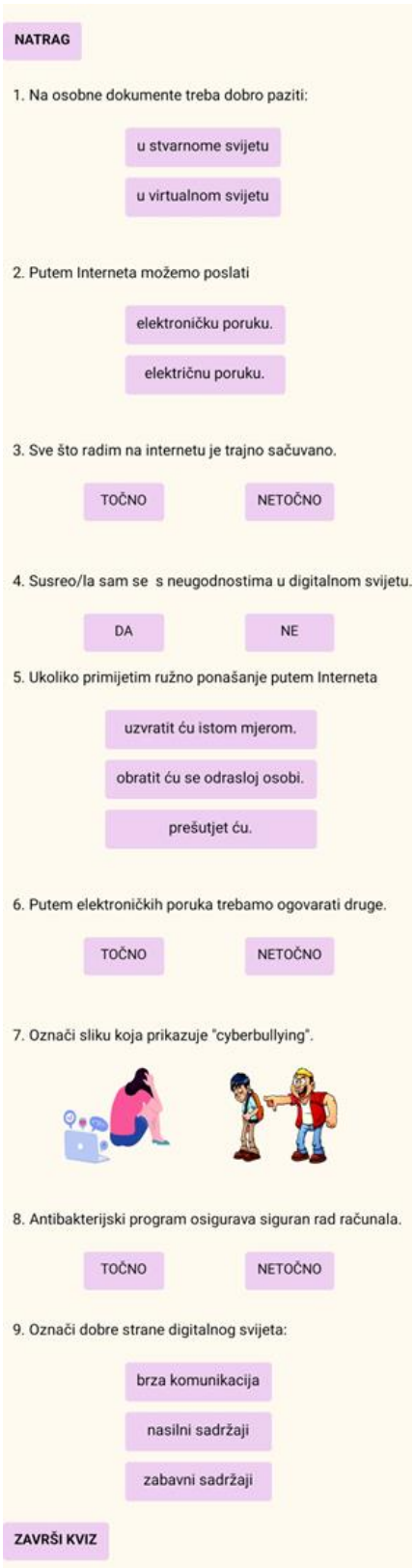
Posljednja cjelina je „Multimedija“ (slika 36) unutar koje učenici ponavljaju što su to digitalni uređaji i čemu služe. Također se utvrđuje što su digitalni sadržaji te kako oni nastaju. Učenici ponavljaju kojim uređajima i programima sami mogu stvarati digitalne sadržaje. Prepoznaju Audacity kao program u kojem mogu uređivati zvuk, Paint u kojem mogu stvarati digitalne crteže te My Photo Creations u kojem mogu stvoriti vlastitu fotoknjigu.

Nakon detaljne razrade svih komponenti koje će se nalaziti u aplikaciji započelo je samo oblikovanje aplikacije. Aplikacija se sastoji od sedam zaslona. Početni zaslon, prikazan na slici 30, sadrži izbornik od šest cjelina. Svaka cjelina predstavlja jedan zaslon, a posljednji zaslon je zaslon s likom žirafe (slika 37) koja učenicima čestita i daje im mogućnost rješavanja nekog drugog kviza. Nakon oblikovanja jednoga kviza, programirao se njegov program, a zatim i testirao, nakon čega se stvarao novi kviz. Jedan kviz predstavlja jedan segment ili modul u modelu vodopada razvoja softvera.

Za potrebe oblikovanja kombinirale su se opcije sučelja MIT App Inventor: *Button, Image, Label, Slider, HorizontalArrangement te TableArrangement.*

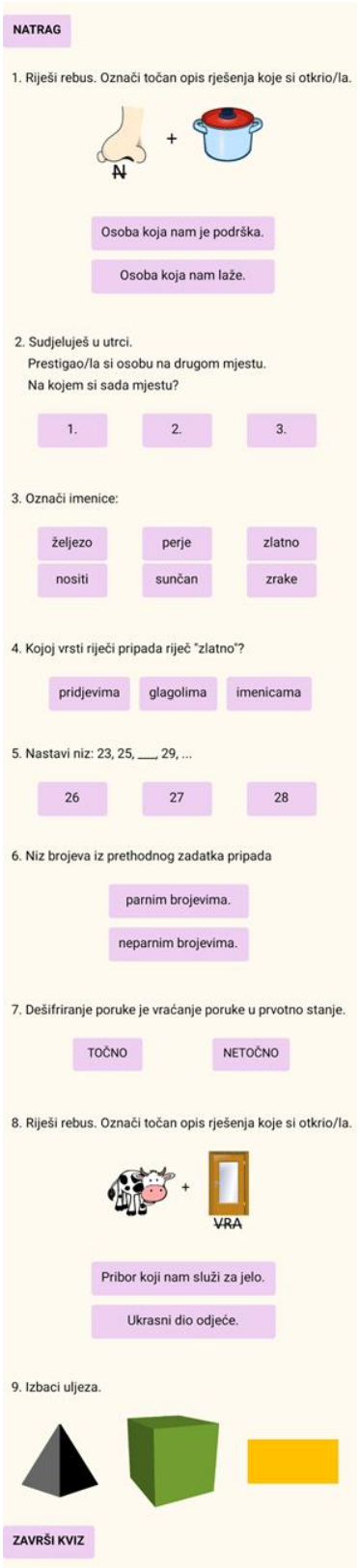
Slika 30. *Početni zaslon aplikacije* Informatika 3

Slika 31. Kviz *Upoznajmo računalo* aplikacije *Informatika 3*



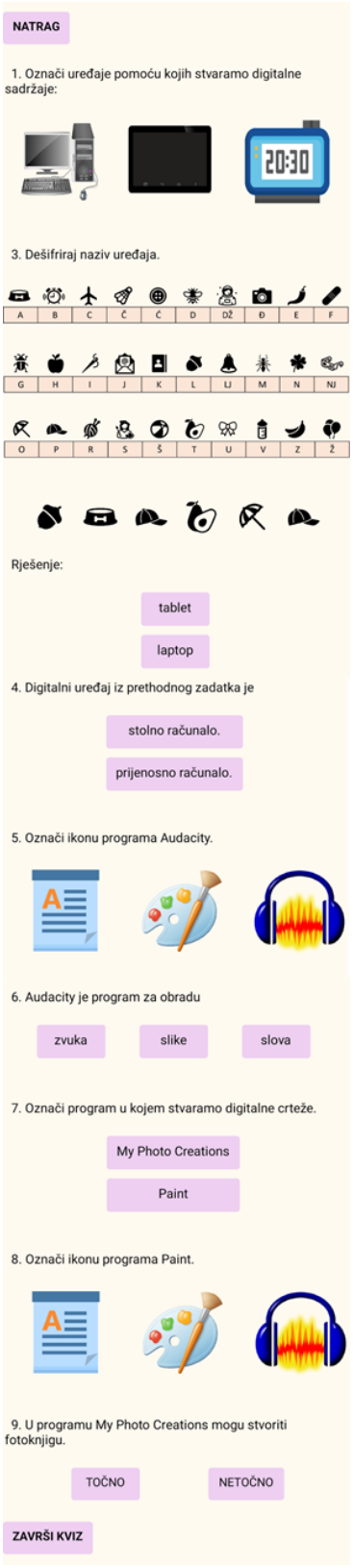
Slika 32. Kviz *e- Svijet* aplikacije *Informatika 3*

Slika 33. Kviz *Stvaranje digitalnog sadržaja* aplikacije *Informatika 3*



Slika 34. Kviz *Rješavanje problema* aplikacije *Informatika 3*

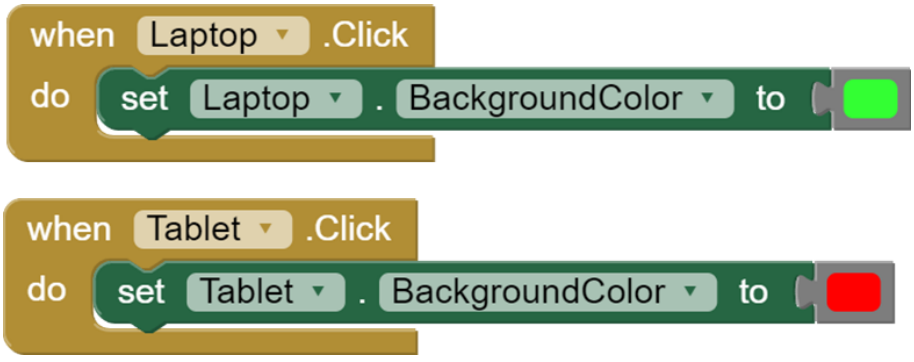
Slika 35. Kviz *Programiranje* aplikacije *Informatika 3*



Slika 36. Kviz *Multimedija* aplikacije *Informatika 3*

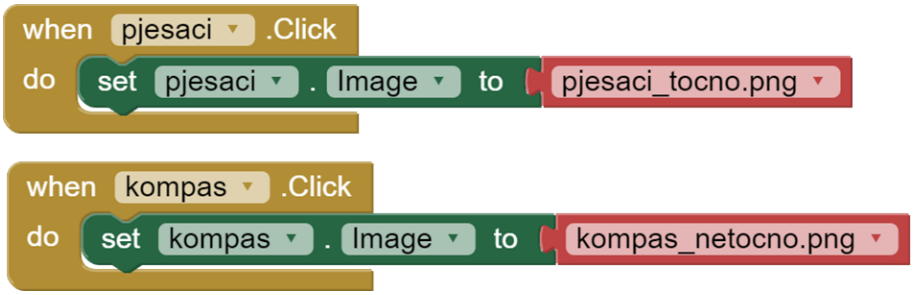
Slika 37. Zaslon prilikom završetka kviza aplikacije *Informatika 3*

Nakon što je aplikacija bila vizualno pripremljena započelo je programiranje u dijelu sučelja gdje se nalaze blokovske naredbe. Odgovori za svaki kviz formirani su na isti način. Odgovori u kojima se odgovara pritiskom na gumb formirani su na način da gumb promijeni boju, ovisno o točnosti, u crvenu ili zelenu.



Slika 38. Primjer programa za promjenu boje gumba u aplikaciji *Informatika 3*

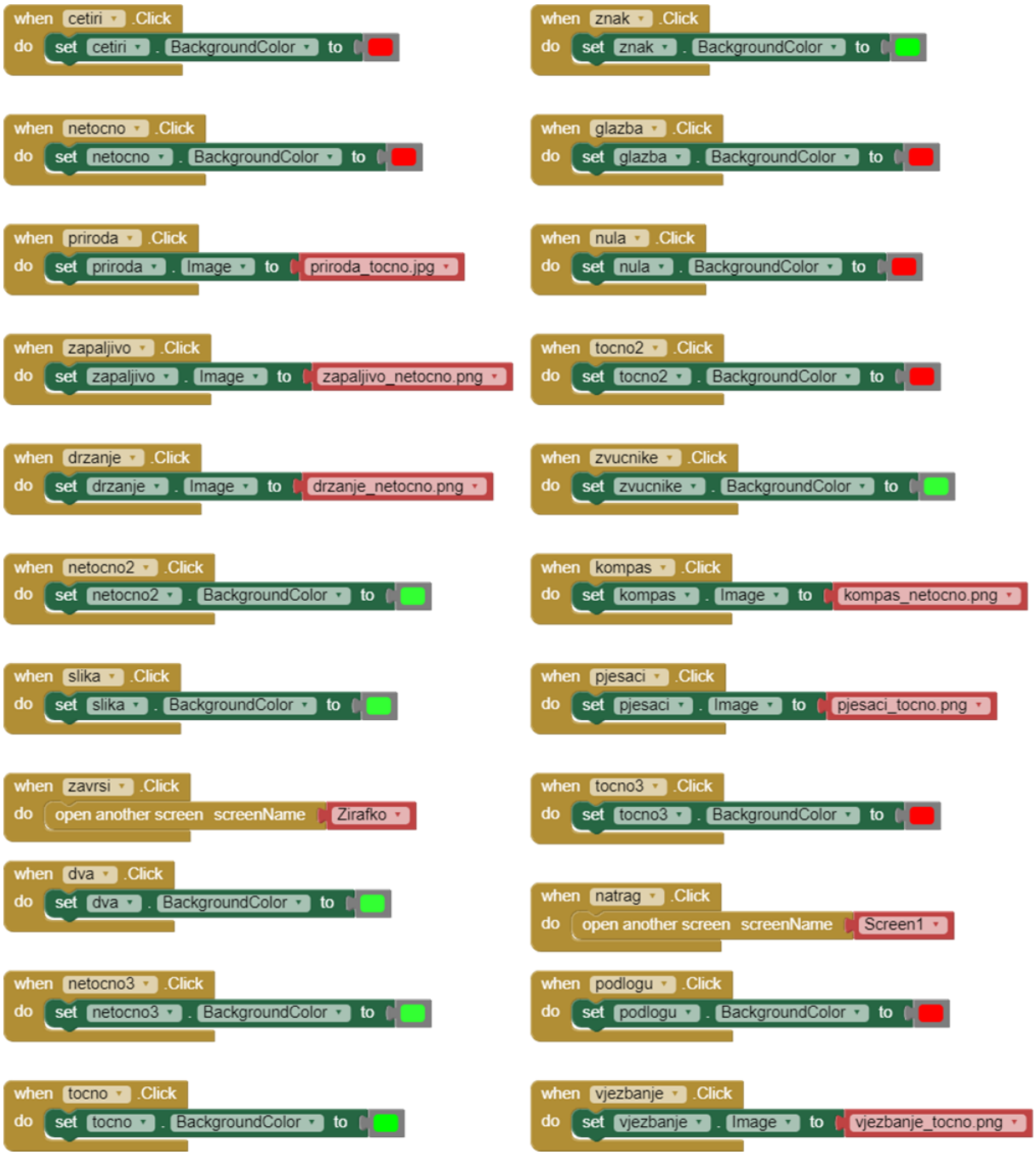
Odgovori koji su ponuđeni kao fotografije ili slike, formirani su na način da se fotografija promijeni u istu tu fotografiju s crvenim ili zelenim obrubom, ovisno o točnosti odgovora.



Slika 39. Primjer programa za promjenu slike u aplikaciji *Informatika 3*

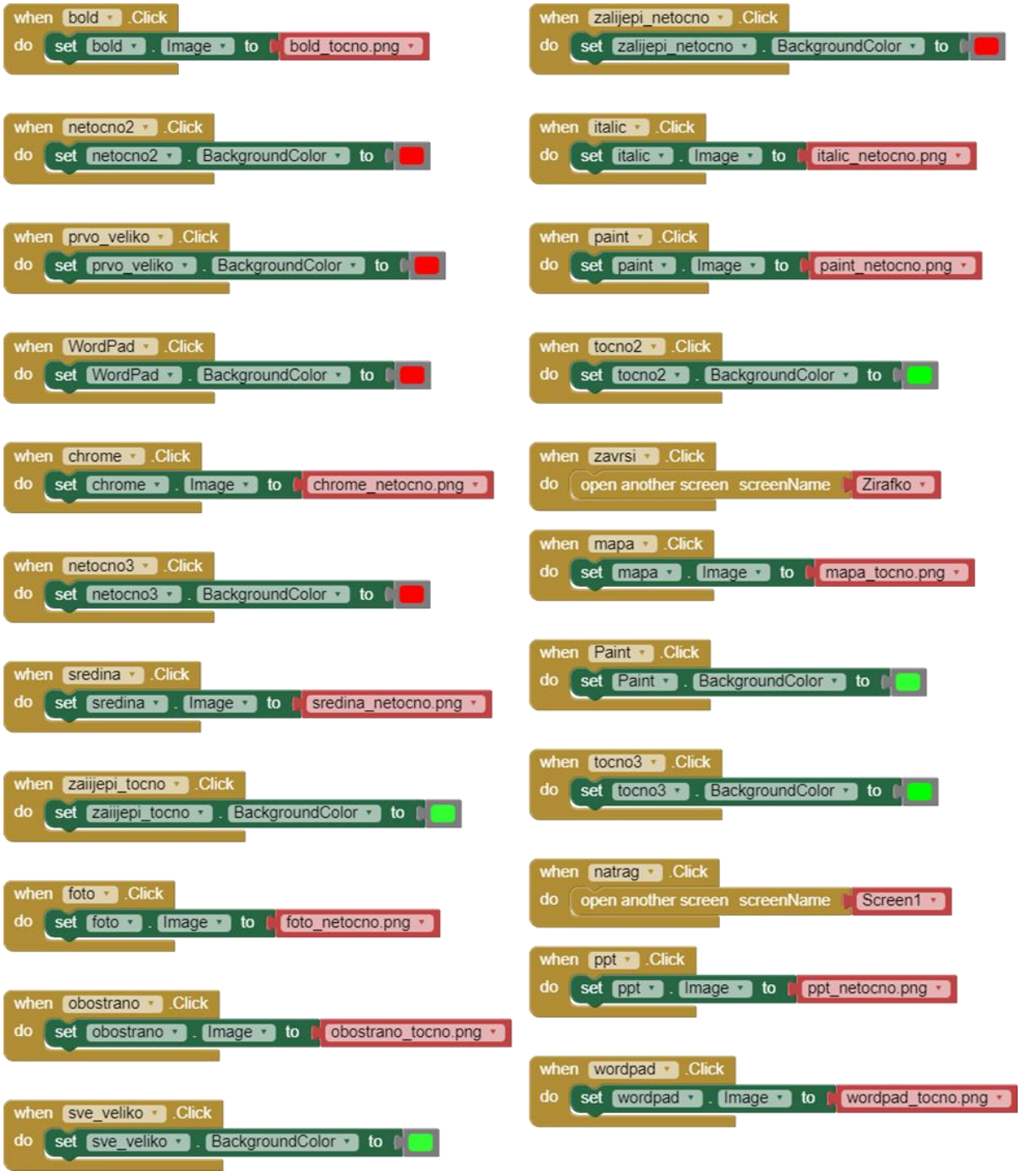
Na dnu svakoga kviza nalazi se gumb *Završi kviz* prilikom čijeg pritiska se pojavljuje lik žirafe koja čestita učenicima na utvrđenom gradivu i daje im mogućnost pritiska na gumb *Dalje* koji ih odvodi na glavni izbornik.

Slika 40. Program početnog zaslona u aplikaciji *Informatika 3*

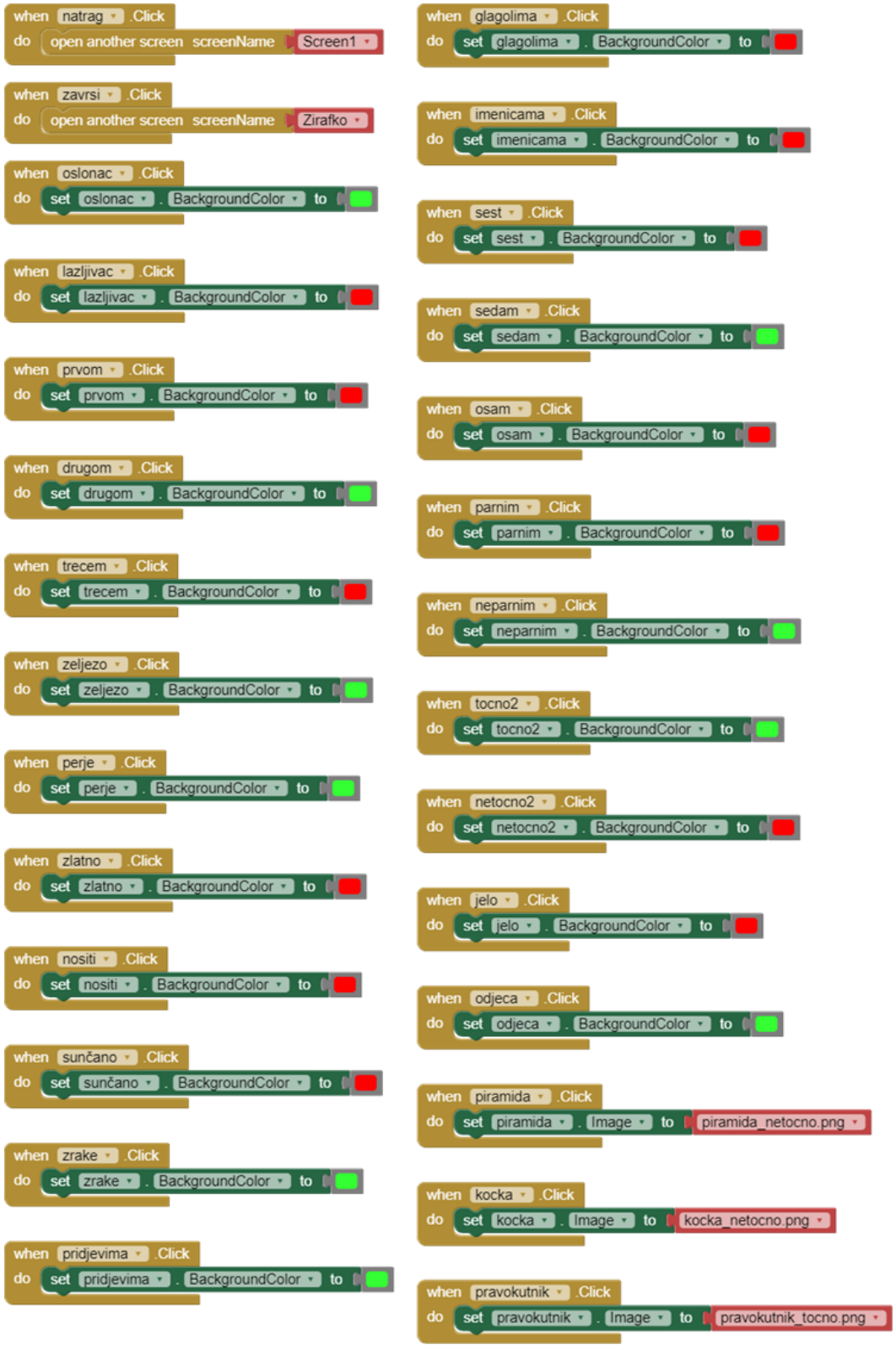


Slika 41. Program kviza *Upoznajmo računalo* aplikaciji *Informatika 3*

Slika 42. Program kviza *e-Svijet* u aplikaciji *Informatika 3*

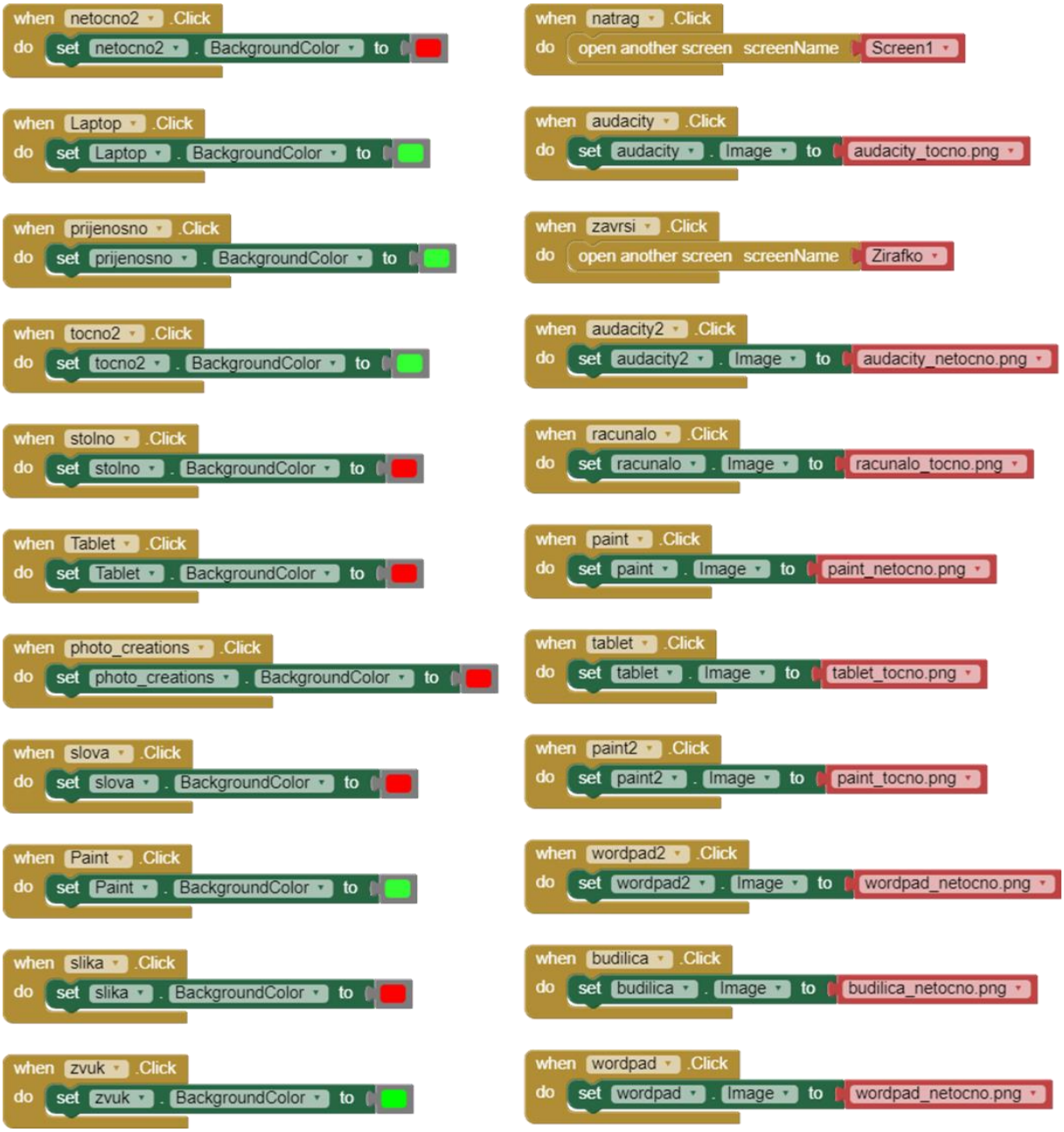


Slika 43. Program kviza *Stvaranje digitalnog sadržaja* u aplikaciji *Informatika 3*

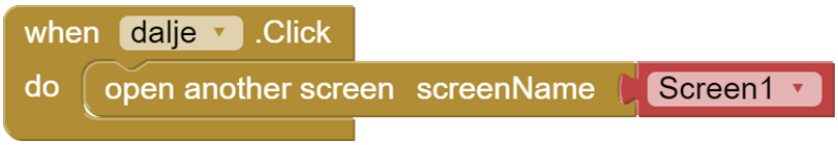


Slika 44. Program kviza *Rješavanje problema* u aplikaciji *Informatika 3*

Slika 45. Program kviza *Programiranje* u aplikaciji *Informatika 3*



Slika 46. Program kviza *Multimedija* u aplikaciji *Informatika 3*



Slika 47. Program završnog zaslona u aplikaciji *Informatika 3*

Nakon programiranja uslijedilo je testiranje aplikacije pomoću mobilne aplikacije MIT AI2 Companion. Kada je rad same aplikacije testiran, aplikacija je puštena u rad i slijedi faza održavanja softvera.

10.4. Mobilna aplikacija „Informatika 3“ u nastavi Informatike

Nastavni sat Informatike održava se kroz sljedeće etape: zamisli, istraži, poveži i izrazi, izradi, pokaži i surađuj. Etape sate nemaju fiksan raspored, već mogu biti raspoređene ovisno o naravi samoga sata. U ovom poglavlju prikazat će se primjer jednog nastavnog sata Informatike u kojem se koristi mobilna aplikacija „Informatika 3“. Nastavni sat bi bio projekt u sklopu cjeline „Rješavanje problema“ iz udžbenika „Moja domena 3“.

Nastavni sat započinje etapom *Zamisli*. Učenici se sklopljenih očiju prisjećaju jednog važnog događaja u svom životu. Razmišljaju o tom događaju te o tome zašto je baš taj događaj njima važan.

Slijedi etapa *Poveži i izrazi*. Učenici pretaču svoje misli u riječi te dijele svoja iskustva s ostatkom razreda. Povezuju taj događaj s njihovim ostalim iskustvima te iznose zašto im je taj događaj važan.

Slijedi etapa *Istraži*. Učenici imaju predložene određene događaje koje oni trebaju poredati od najranijeg prema posljednjem. Učenici istražuju i promišljaju o tome koji su se događaji odvili prije, a koji kasnije. Ukoliko imaju potrebe učenici imaju mogućnost komunikacije u digitalnom okruženju s roditeljima kako bi im pomogli poredati neke događaje. Događaji koji su predloženi učenicima: rođenje učenika, rođenje braće/sestara, polazak u vrtić, polazak u školu, izbijanje prvog zubića, prvi izgubljeni zubić, prva simpatija, prvi kućni ljubimac te prvo sportsko natjecanje. Učenici mogu odabrati nekoliko događaja koje će poredati.

Nakon istraživanja započinje etapa *Izradi*. Učenici svakom događaju koji su odlučili iskoristiti dodjeljuju njegov simbol. Zatim izrađuju svoje lente vremena koristeći osmišljene simbole.

U etapi *Pokaži* učenici prezentiraju svoje radove ostatku razreda. Ostali učenici pretpostavljaju za koji je događaj vezan simbol na lenti vremena, a učenik predstavlja koje je događaje odabrao prikazati na lenti vremena.

Za kraj ovoga projekta provodi se etapa *Surađuj*. Učenici u sklopu mobilne aplikacije „Informatika 3“ rješavaju kviz cjeline „Rješavanje problema“ koji prikazuje slika 34. Ova se etapa provodi radom u paru te njome učenici zaokružuju jednu naučenu cjelinu gradiva iz udžbenika „Moja domena 3“.

Prikazan je jedan primjer sata u kojem se koristi aplikacija „Informatika 3“ za ponavljanje gradiva i zaokruživanja jedne naučene cjeline nakon provođenja projekta. Aplikacija „Informatika“ 3 može se koristiti u bilo kojoj etapi sata Informatike, kao i aktivnost za domaću zadaću. Aplikacija može poslužiti kao motivacijska aktivnost ili se iskoristiti u glavnom dijelu sata. Prilikom korištenja aplikacije učenici mogu surađivati u paru ili skupini, a mogu i samostalno rješavati kviz. Bilo bi poželjno provjeriti rješenja na kraju odigranoga kviza te komentirati odgovore koji mogu potaknuti novi tijek razgovora.

11. ZAKLJUČAK

Kroz rad opisana je važnost učenja i poučavanja predmeta Informatike za suvremeno društvo. Nove generacije učenika sve su izloženije informacijsko-komunikacijskoj tehnologiji, a učitelji imaju važnu ulogu u poučavanju učenika kako se njome koristiti na siguran način. Informacijsko-komunikacijska tehnologija uvelike je zastupljena u obrazovanju, a kroz vrijeme će biti još zastupljenija. Ona donosi brojne dobrobiti u nastavnom procesu, od same motivacije učenika za rad, učenja kroz igru, pa sve do stvaranja programa i jednostavnih igara od najranijeg osnovnoškolskog obrazovanja. U razrednoj nastavi osnovne škole učenici uče kako se sigurno koristiti digitalnim uređajima, čemu oni služe, kako njima stvarati sadržaje, ali i razvijaju apstraktno i računalno razmišljanje te započinju svoje prve korake u programiranju. Odlično okruženje koje služi programiranju djeci najmlađe dobi je Scratch koji pomoću blokova naredbi učenicima omogućava vizualizaciju apstraktnog procesa programiranja. U Scratchu učenici mogu stvarati jednostavne programe i igre te razvijati svoje računalno razmišljanje na zabavan i igrolik način. Učenicima programiranje približava i vrlo korisna u nastavi organizacija Code.org.

Kroz rad je spomenuto i učenje kroz igru te izražavanje kreativnosti učenika, a to učitelji mogu potražiti u brojnim edukativnim aplikacijama koje se nude na tržištu. Svakodnevno korištenje mobilnih aplikacija danas je jedna uopćena stvar, a one dakako mogu poslužiti i u edukativne svrhe. Štoviše, postoji programsko okruženje slično Scratchu koje nudi mogućnost besplatnog razvoja mobilnih aplikacija. Vizualno-blokovsko programsko okruženje MIT App Inventor može poslužiti kao korak dalje od programiranja u Scratchu, a u ovom radu predstavljene su i dvije edukativne mobilne aplikacije razvijene upravo u njemu. Programiranje u MIT App Inventoru dostupno je učiteljima koji mogu stvarati vlastite aplikacije, ali i učenicima koji mogu u njemu istraživati i stvoriti jednostavne softvere. Aplikacije izrađene za potrebe ovoga rada su mobilna aplikacija „Kalendar prirode“ te mobilna aplikacija „Informatika 3“. Prikazano je kako se edukativne mobilne aplikacije ne moraju koristiti samo na satu Informatike, već i u drugim nastavnim predmetima, kao što je u ovom slučaju predmet Priroda i društvo. Cilj ovoga rada je pokazati kako se programiranjem u MIT App Inventoru može baviti bilo tko, od učenika razredne nastave do odraslih te kako se u njemu mogu razvijati vrlo korisne aplikacije različite složenosti. Kao cilj rada, istaknula bi se i potreba da se informacijsko-komunikacijska tehnologija koristi u redovnoj nastavi u određenoj mjeri kako bi učenici bili odgovorni korisnici, ali i stvaratelji iste.

12. LITERATURA

Begel, A., & Klopfer, E. (2007). Starlogo TNG: An introduction to game development. Journal of E-Learning, 53, 146. Pristup 14.05.2023. https://www.academia.edu/50960444/Starlogo_TNG_An_Introduction_to_Game_Development .

Code.org. About Us. Pristup 13.05.2023. https://code.org/about .

Code.org (2017). What's the impact of the Hour of Code. Pristup 13.05.2023. https://code.org/about/evaluation/hourofcode .

De Zan, I. (2005). Metodika nastave prirode i društva, Zagreb: Školska knjiga

Dumančić, M. Sadržaji kolegija Metodike informatike 1 i Metodike Informatike 2

Kafai, Y. B. (2016). From computational thinking to computational participation in K-12 education. Communications of the ACM, 59(8), 26–27. Pristup 14.05.2023. https://www.academia.edu/27424884/Education_From_Computational_Thinking_to_Computational_Participation_in_K_12_Education .

Letina, A. Sadržaji kolegija Metodike prirode i društva 1, Metodike prirode i društva 2, Metodike prirode i društva 3

Maloney, J., Resnick,M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. ACM Transactions on Computing Education (TOCE), 10(4), 16. Pristup 13.05.2023. https://web.media.mit.edu/~jmaloney/papers/ScratchLangAndEnvironment.pdf .

Ministarstvo znanosti i obrazovanja (2018). Kurikulum nastavnog predmeta Informatike za osnovne škole i gimnazije u Republici Hrvatskoj. Preuzeto 20.04.2023. s https://narodne-novine.nn.hr/clanci/sluzbeni/2018_03_22_436.html .

Ministarstvo znanosti i obrazovanja (2019). Kurikulum za nastavni predmet Priroda i društvo za osnovne škole u Republici Hrvatskoj. Preuzeto 01.04.2023. s https://narodne-novine.nn.hr/clanci/sluzbeni/2019_01_7_147.html .

Mladenović, M. (2019). Poučavanje početnog programiranja oblikovanjem računalnih igara Pristup 15.05.2023. https://dabar.srce.hr/islandora/object/pmfst:760 .

Marji, M. (2014). Learn to program with Scratch. No Starch Press

Oreški, P. Sadržaji kolegija Programiranje

Patton, E. W., Tissenbaum, M., Harunani, F. (2019). MIT App Inventor: Objectives, Design, and Development, 3, 31-49.

Podrug, I. (2017). Mogućnosti primjene mobilnih aplikacija u nastavi prirode i biologije. *Educatio biologiae,* 3, 165-176. Preuzeto 15.05.2023. s https://hrcak.srce.hr/192688 .

Rihter, B., Toić Dlačić, K. (2020*). Moja domena 3, udžbenik iz informatike za treći razred osnovne škole*. Zagreb: Alfa.

Tomljenović, K. (2018). Računalno razmišljanje i uloga učenja pomoću igre na njegov razvoj. Učiteljski fakultet. Preuzeto 15.05.2023. s

https://scholar.google.ae/citations?view_op=view_citation&hl=en&user=FXkLI0kAAAAJ&citation_for_view=FXkLI0kAAAAJ:qjMakFHDy7sC .

SCRATCH. About Scratch. Pristup 13. 05. 2023. https://scratch.mit.edu/about .

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Wilson, A. i Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming.

Wolber, D., Abelson, H., Spertus, E., Looney, L. (2014). App Inventor 2: Create your own Android Apps. O'Reilly Media, Inc. Pristup 27. 03. 2023. http://www.appinventor.org/book2 .

13. PRILOZI I DODACI

Slika 1. Vizualno-blokovsko programsko okruženje Scratch 8

Slika 2. Softverski proces prema modelu vodopada (Manger, 2016, 6) 14

Slika 3. Softverski proces prema modelu evolucijskog razvoja (Manger, 2016, 7) 15

Slika 4. Softverski proces prema modelu inkrementalnog razvoja (Manger, 2016, 8) 16

Slika 5. Softverski proces prema modelu formalnog razvoja (Manger, 2016, 9) 16

Slika 6. Softverski proces prema modelu grafičkoga razvoja (Manger, 2016, 10) 17

Slika 7. Softverski proces prema modelu usmjerenom na ponovnu uporabu (Manger, 2016, 11)17

Slika 8. MIT App Inventor: Designer 20

Slika 9. MIT App Inventor: Blocks 21

Slika 10. Glavni izbornik aplikacije Kalendar prirode 22

Slika 11. Obrazac za odabir nadnevka u aplikaciji Kalendar prirode 23

Slika 12. Opcija DatePicker u aplikaciji Kalendar prirode 23

Slika 13. Primjer opcije Spinner u aplikaciji Kalendar prirode 24

Slika 14. Primjer opcije Slider u aplikaciji Kalendar prirode 24

Slika 15. Primjer izbornika za odabir dana u aplikaciji Kalendar prirode 25

Slika 16. Prikaz obrasca jednoga dana nizinskoga zavičaja aplikacije Kalendar prirode..... 26

Slika 17. Prikaz obrasca jednoga dana brežuljkastoga zavičaja aplikacije Kalendar prirode 27

Slika 18. Prikaz obrasca jednoga dana gorskoga zavičaja aplikacije Kalendar prirode 28

Slika 19. Opcija CheckBox u aplikaciji Kalendar prirode 29

Slika 20. Opcija TextBox u aplikaciji Kalendar prirode 30

Slika 21. Prikaz obrasca jednoga dana primorskoga zavičaja aplikacije Kalendar prirode..... 30

Slika 22. Prikaz programa početnog zaslona aplikacije Kalendar prirode 34

Slika 23. Prikaz programa za ispunjavanje nadnevka aplikacije Kalendar prirode 34

Slika 24. Globalne liste aplikacije Kalendar prirode 35
Slika 25. Primjer programa za povratak na izbornik Odabir dana 36
Slika 26. Primjer programa nizinskoga zavičaja za spremanje podataka u TinyDB 37
Slika 27. Primjer programa brežuljkastoga zavičaja za spremanje podataka u TinyDB 38
Slika 28. Primjer programa gorskoga zavičaja za spremanje podataka u TinyDB 39
Slika 29. Primjer programa gorskoga zavičaja za spremanje podataka u TinyDB 40
Slika 30. Početni zaslon aplikacije Informatika 3 44
Slika 31. Kviz Upoznajmo računalo aplikacije Informatika 3 45
Slika 32. Kviz e- Svijet aplikacije Informatika 3 46
Slika 33. Kviz Stvaranje digitalnog sadržaja aplikacije Informatika 3 47
Slika 34. Kviz Rješavanje problema aplikacije Informatika 3 48
Slika 35. Kviz Programiranje aplikacije Informatika 3 49
Slika 36. Kviz Multimedija aplikacije Informatika 3 50
Slika 37. Zaslon prilikom završetka kviza aplikacije Informatika 3 51
Slika 38. Primjer programa za promjenu boje gumba u aplikaciji Informatika 3 51
Slika 39. Primjer programa za promjenu slike u aplikaciji Informatika 3 52
Slika 40. Program početnog zaslona u aplikaciji Informatika 3 52
Slika 41. Program kviza Upoznajmo računalo aplikaciji Informatika 3 53
Slika 42. Program kviza e-Svijet u aplikaciji Informatika 3 54
Slika 43. Program kviza Stvaranje digitalnog sadržaja u aplikaciji Informatika 3 55
Slika 44. Program kviza Rješavanje problema u aplikaciji Informatika 3 56
Slika 45. Program kviza Programiranje u aplikaciji Informatika 3 57
Slika 46. Program kviza Multimedija u aplikaciji Informatika 3 58
Slika 47. Program završnog zaslona u aplikaciji Informatika 3 58

14. IZJAVA O IZVORNOSTI RADA

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristila drugim izvorima osim onih koji su u njemu navedeni.

_____

(vlastoručni potpis studenta)