Didaktičke igre za razvoj računalnog razmišljanja

Anđelić, Matea

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:195:807932

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository

zir.nsk.hr

DIGITALNA KNJIŽNICA

Image not found or type unknown

Sveučilište u Rijeci – Odjel za informatiku

Informacijski i komunikacijski sustavi

Matea Anđelić

Didaktičke igre za razvoj računalnog razmišljanja

Diplomski rad

Mentor: prof. dr. sc. Nataša Hoić-Božić

Rijeka, prosinac 2018.

Sažetak

Danas se sve češće u školama u sadržaj informatike uključuju didaktičke igre. Kroz računalne igre učenici usvajaju koncepte programiranja i koncepte računalnog razmišljanja. Usvajanje koncepata programiranja se odnosi na usvajanje slijedova koraka, varijabli, petlji i uvjeta. Usvajanje koncepata računalnog razmišljanja podrazumijeva usvajanje dekompozicije, prepoznavanja uzoraka, apstrakcije i algoritama.

Računalno razmišljanje i programiranje važni su elementi informatike no istovremeno su učenicima izazovni za usvajanje.

Ključne riječi: Računalne igre, didaktičke igre, računalno razmišljanje, programiranje, slijedovi koraka, varijable, petlje, uvjeti, dekompozicija, prepoznavanje uzoraka, apstrakcija, algoritmi

Sadržaj

1.Uvod 4

2. Računalno razmišljanje 5

2.1. Definicija 5

2.2. Tehnike računalnog razmišljanja 6

2.3. Važnost računalnog razmišljanja 11

3.Računalne igre u edukaciji 12

3.1.Usvojene vještine primjenom računalnih igara u edukaciji 13

3.1.1.Općenite vještine 13

3.1.2.Vještine programiranja 13

4.Klasifikacija igara 15

4.1.Usvajanje koncepata programiranja 15

4.1.1.Usvajanje slijedova koraka 15

4.1.2.Usvajanje varijabli 19

4.1.3.Usvajanje petlji 24

4.1.4.Usvajanje uvjeta 28

4.2.Usvajanje koncepata računalnog razmišljanja 31

4.2.1.Usvajanje dekompozicije 31

4.2.2.Usvajanje prepoznavanja uzoraka 33

4.2.3.Usvajanje apstrakcije 35

4.2.4.Usvajanje algoritama 37

5.Primjeri zadataka za učenike 42

6.Zaključak 46

7.Popis literature 47

8.Popis slika 49

1.Uvod

Velike količine podataka i brojne nove tehnološke inovacije su ono što danas susrećemo u svakodnevnom životu. Stoga, bitan element današnjice su informacijska i komunikacijska tehnologija (ICT) i sustavi. Temelj razvoja bilo kojeg programa je znanje. Znanje se postiže dobro strukturiranim obrazovanjem, pa možemo reći da se uspješnost razvoja tehnologije temelji upravo na obrazovanju.

Važnost informatike u obrazovanju posljednjih godina prepoznata je u svim školskim sustavima, kao način poboljšanja produktivnijeg rada s računalima kod djece. Koliko je bitno programiranje u informatici bi možda trebalo još više naglasiti jer se kod djeteta tako razvija logički način razmišljanja koji pridonosi kreativnom rješavanju problema, te ih na neki način priprema za budućnost razvijajući brojne vještine kao što su upornost, uspješan timski rad, vježbanje načina razmišljanja, pedantnost (posvećenost detaljima) i slično[1].

Programiranje, kao važan element informatike, je i zahtjevan element, a često i nedovoljno zanimljiv učenicima. Način na koji je moguće sadržaj informatike učiniti zanimljivijim su didaktičke igre, koje je potrebno uključiti kod učenika mlađe dobi, kako bi se već u osnovnoj školi kod učenika razvijalo računalno, algoritamsko razmišljanje. Razvoj takvog načina razmišljanja postiže se brojnim računalnim didaktičkim igricama koje uključuju igre pamćenja, slagalice, mozgalice, zagonetke i slično. Kroz ovakve računalne igrice učenici osnovne škole razvijaju vještine potrebne za programiranje ali i općenito digitalne vještine.

Cilj diplomskog rada je klasifikacija igara i zadataka na način da se odnose na usvajanje novih koncepata u programiranju, te izrada primjera didaktičkih igara za razvoj računalnog razmišljanja i koncepata programiranja kod učenika mlađe dobi.

2. Računalno razmišljanje

2.1. Definicija

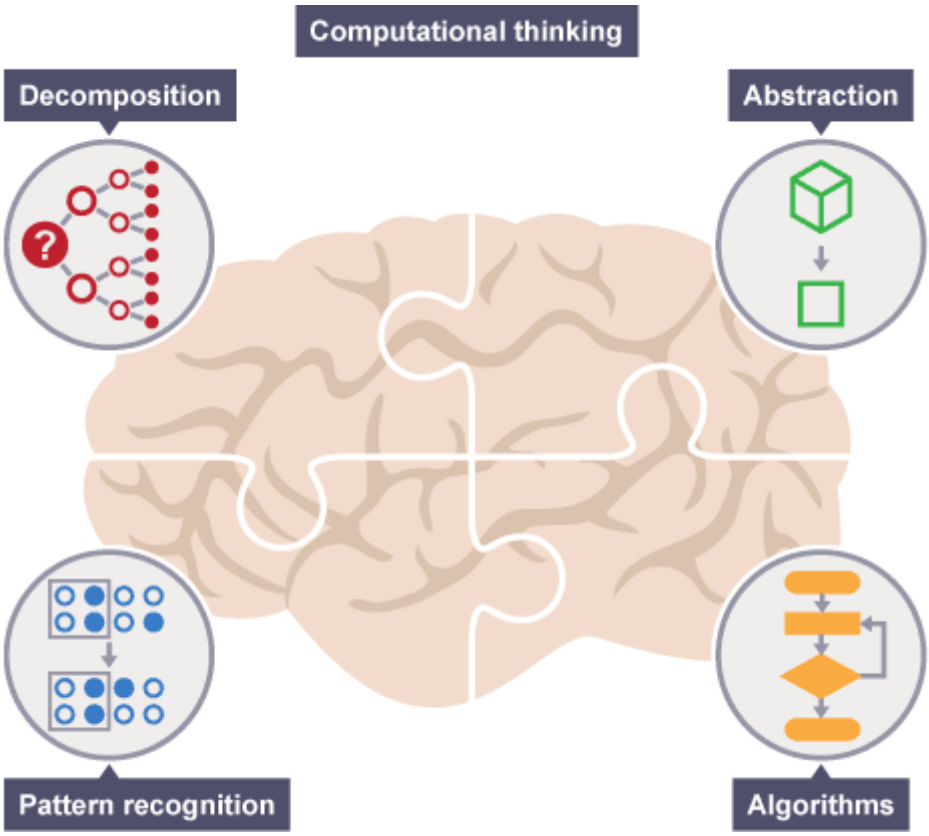
Računala se mogu koristiti za rješavanje brojnih problema. Međutim, prije nego što se neki problem može riješiti, potrebno je razumjeti sam problem, kao i načine na koje se može riješiti. Računalno razmišljanje nam omogućuje da to učinimo.

Računalno razmišljanje je proces razmišljanja uključen u formuliranje problema i izražavanje njegovog rješenja na takav način da bude strojno razumljiv i da se može strojno izvršiti[3]. Dakle, omogućuje nam da za neki složeni problem shvatimo što je uopće problem i da se pri tome razviju moguća rješenja, koja se mogu predstaviti na način da ih računalo, čovjek ili oboje mogu razumjeti.

Povijest računalnog razmišljanja započinje pedesetih godina prošlog stoljeća, a sami pojam računalnog razmišljanja je prvi puta bio upotrijebljen 1980.godine. Računalno razmišljanje se koristi kod algoritamskog rješavanja kompleksnih problema. Osoba koja ima sposobnost računalnog razmišljanja je sposobna logički organizirati podatke i analizirati podatke. Sposobna je razbiti problem na manje dijelove, približiti problem pomoću nekih programskih tehnika kao što su iteracije, logičke operacije i slično. Osim toga, zna identificirati, analizirati i primijeniti moguća rješenja s ciljem postizanja učinkovitog rješavanja problema[3].

2.2. Tehnike računalnog razmišljanja

Četiri su temeljne tehnike računalnog razmišljanja[4]. Prikazane su na slici 1. To su: dekompozicija, prepoznavanje uzoraka, apstrakcija i algoritmi.



Slika 1. Temeljne tehnike računalnog razmišljanja

Dekompozicija (engl.decomposition)

Dekompozicija je razbijanje složenog problema ili sustava u manje probleme. Složen problem je onaj kojeg na prvi pogled ne znamo kako lako riješiti. Nakon razbijanja složenog problema u manje probleme, ti manji problemi mogu biti lakše razumljivi i lakši za rješavanje. Na taj način je omogućeno bolje razumijevanje novih situacija i lakše dizajniranje velikih sustava. Primjerice problem pravljenja doručka se može razbiti u manje zasebne probleme kao što su primjerice napraviti tost, napraviti čaj, skuhati jaje i slično. Svaka od ovih aktivnosti se može dodatno podijeliti u niz koraka. Dakle, kod dekompozicije svaki se problem može razbiti na više dijelova koji će se integrirati kasnije u procesu. Primjerice kod razvoja igre različiti ljudi

mogu samostalno izrađivati i stvarati različite dijelove igre pod uvjetom da su ključni aspekti prethodno dogovoreni, a na kraju se ti dijelovi integriraju. U školama se neka nastavna cjelina može podijeliti na više dijelova kako bi se lakše radilo s manjim dijelovima, a na taj način je i učenicima lakše učiti. Dekompozicija se u školama još može primjeniti podjelom složenog problema na manje dijelove i to na način da se ti manji problemi rješavaju na isti način (rekurzivno)[17].

Prepoznavanje uzoraka (engl.pattern recognition)

Nakon dekompozicije, odnosno razbijanja problema na manje probleme, svaki od tih problema se može pogledati pojedinačno i u ovom koraku se traže sličnosti između manjih problema. Traže se sličnosti i vezi između problema, a kad se pronađu postavlja se pitanje kako najbolje iskoristiti ta saznanja. To je način brzog rješavanja novih problema temeljenih na prethodnim rješenjima problema i na temelju prethodnog iskustva. Rješenja nekih specifičnih problema se mogu prilagoditi na način da se riješi cijela klasa sličnih problema. Tada, kad god se susreće problem te vrste, može se primjeniti opće rješenje. U školama se prepoznavanja uzoraka može primjeniti na način da učenici traže sličnosti među nekim zadacima i zajedničke elemente u tim zadacima primjerice, nakon toga pronalaze rješenje zadatka koje bi bilo prilagodljivo svim ostalim zadacima[17]. Konkretan primjer u informatici u školama možemo uočiti primjerice kod pretvorbe brojeva iz jednog brojevnog sustava u drugi. Kada učenik uoči točno pravilo rješavanja jedne vrste problema pretvorbe brojeva, znat će to pravilo primjeniti na svaki idući zadatak tog tipa.

Apstrakcija (engl.abstraction)

Apstrakcija olakšava razmišljanje o problemima ili sustavima. Apstrakcija podrazumijeva fokusiranje samo na važne informacije, ignorirajući nevažne detalje. Ona podrazumijeva proces stvaranja nekog problema razumljivijim otklanjajući nepotrebne pojedinosti. U apstrakciji je važmo znati odabrati prave detalje koje treba zanemariti kako bi problem postao lakši, a da se pri tome ne izgubi ono što je važno. U školama se apstrakcija može primjeniti na način da se smanji složenost nekog gradiva uklanjanjem nepotrebnih detalja, da se odabere način kako predstaviti to gradivo i kako manipulirati njime na koristan način. Kod nekih kompleksnih gradiva se može koristiti strukturiranje gradiva, u svrhu skrivanja kompleksnosti tog gradiva, te treba znati prepoznati odnose između apstrakcija[17]. Primjer primjene apstrakcije u informatici može biti učenje sklopovlja računala. Učenje primjerice samo ulaznih, izlaznih dijelova ili možda samo vanjskih, unutarnjih dijelova računala, učenici vide

kao apstrakciju, jer se fokusiraju recimo samo na unutarnje dijelove računala, zanemarujući ostale detalje, odnosno vanjske dijelove računala.

Algoritmi (engl.algorithms)

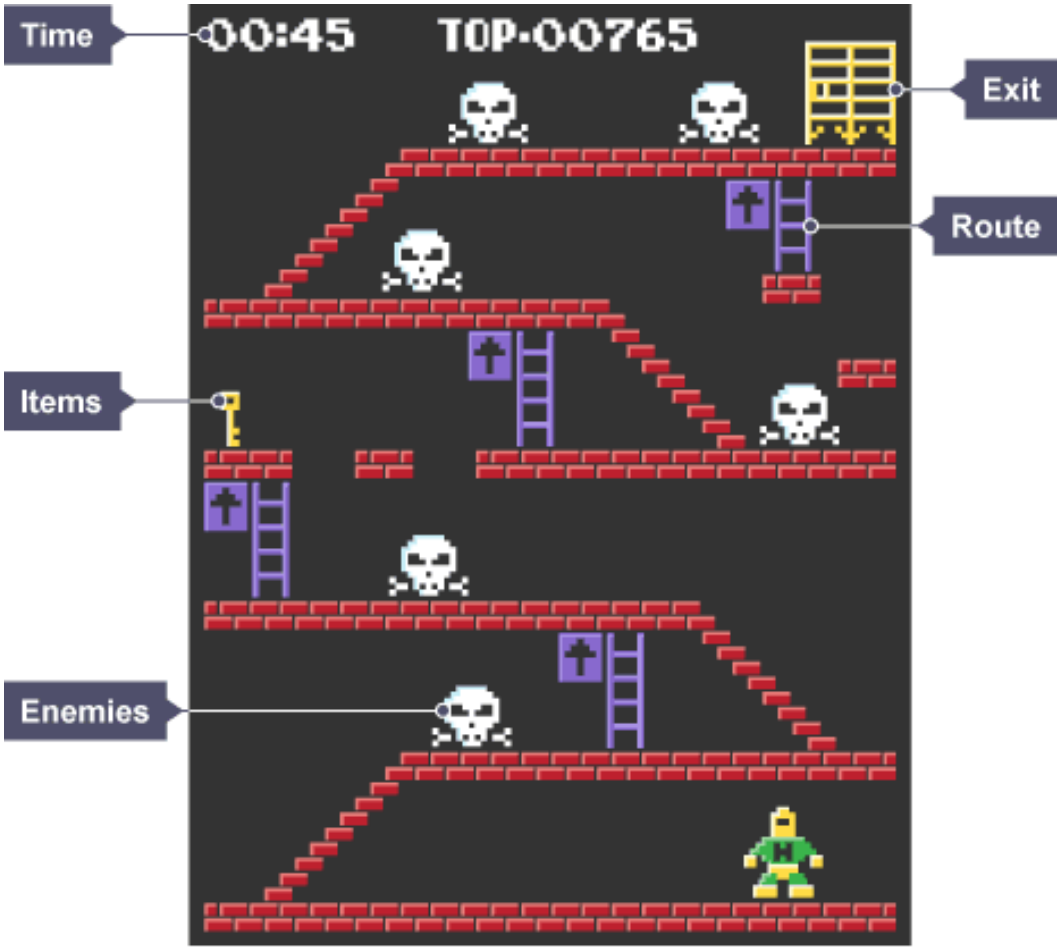
Algoritmi podrazumijevaju razvijanje rješenja problema korak po korak, ili pravila koja vode ka rješenju problema, dakle oblikuju se jednostavni koraci ili pravila za rješavanje svakog od problema. Neki problemi su jednokratni, osmisli se rješenje za njega, primjeni se to rješenje i rješava se sljedeći problem. Ali u slučaju kada imamo slične probleme koji se iznova pojavljuju više puta, tada je potrebno osmisliti neki algoritam, odnosno rješenje koje radi svaki put i u tom slučaju se ne mora svaki put iznova razmišljati. Primjer za algoritam u školstvu je učenje množenja i dijeljenja. Ako se točno prate pravila, može se pronaći rješenje za bilo koji zadatak množenja i dijeljenja. Jednom kad se algoritam shvati, ne mora se ići od nule za svaki novi sljedeći problem tog tipa. Vještinu razvijanja algoritama učenici razvijaju kroz pisanje vlastitih računalnih programa. Primjena algoritama u školama može obuhvaćati: formuliranje uputa za postizanje željenog učinka, uputa koje treba slijediti u određenom redoslijedu (slijed koraka), uputa koje koriste aritmetičke i logičke operacije, upute za pohranu i manipulaciju podacima (varijable), za ponavljanje nekog dijela (petlje) i slično[17].

Svaka od prethodno navedene četiri tehnike je jednako važna kao i ostale, te ispravna primjena sve četiri tehnike dovodi do rješavanja problema. Ako jedna od njih nedostaje, problem neće moći biti uspješno riješen.

Računalno razmišljanje nije programiranje. Nije čak niti razmišljanje poput računala, jer računala, naravno, ne mogu razmišljati. Jednostavno rečeno, programiranjem govorimo računalu što učiniti i kako to učiniti, a računalno razmišljanje omogućuje da se točno odredi što će se reći računalu. Najjednostavniji primjer koji objašnjava vezu računalnog razmišljanja i programiranja je recimo odlazak na neko mjesto gdje nikada prije nismo bili. Vrlo vjerojatno je da bismo planirali svoju rutu prije nego izađemo iz kuće, odnosno razmotrili bismo dostupne rute i koje su rute najbolje – to može biti najkraći put, najbrži put ili slično. Slijedili bismo upute kako doći do odredišta korak po korak. U ovom slučaju planiranje rute je poput računalnog razmišljanja, a praćenje uputa je poput programiranja[4].

Biti u stanju pretvoriti složeni problem u jedan koji možemo lako razumjeti je vještina koja je izuzetno korisna. Dan je primjer igranja računalne igre. Da bi se dovršila neka razina u igri, postoje neke stavke u igri koje treba znati. Slika 2 prikazuje ovaj navedeni primjer, gdje su

istaknute neke stavke. Primjer zapravo objašnjava gdje se računalno razmišljanje koristi za rješavanje složenog problema. Ako pogledamo sliku, vidimo da je potrebno primjerice znati koje stvari (items) treba prikupiti, kako ih se može prikupiti i koliko je vremena još ostalo za prikupljanje istih (time). Nadalje, treba znati gdje je izlaz (exit) i koja je najbolja ruta (route) za postizanje cilja u što kraćem vremenu, kakvi neprijatelji (enemies) u igrici postoje i njihove slabe točke. Kada znamo ove detalje, kako je izraditi strategiju za dovršavanje razine u igrici na najučinkovitiji način.



Slika 2. Primjer računalnog razmišljanja za rješavanje složenog problema

I kod stvaranja vlastite računalne igre, to su upravo tipovi pitanja o kojima bi se trebalo razmišljati i odgovarati prije programiranja igre. Iz ovog primjera računalnog razmišljanja za rješavanje složenog problema se može zaključiti da je svaki složeni problem bio podijeljen u nekoliko malih odluka i koraka (npr.gdje ići i kako dovršiti razinu - **dekompozicija**), fokus je

bio samo na nekim detaljima (npr.na vremenu ili mjestu izlaza - **apstrakcija**), poznaju se prethodni slični problemi (**prepoznavanje uzoraka**) kako bi se izradio plan djelovanja korak po korak (**algoritmi**)[4].

2.3. Važnost računalnog razmišljanja

Kroz prethodno navedeni primjer je očigledno da se igranjem računalnih igara vrlo lako može razviti računalni način razmišljanja, stoga je velika važnost uključivanja ovakvih igara u edukaciju u osnovnim školama.

Rješavanje problema, analiza valjanosti rješenja, razna uočavanja u podacima, kreativnost - to su sve osnovne vještine za radna mjesta, stoga je veoma bitno podučavati učenike ranije dobi na način da razvijaju računalno razmišljanje, što će doprinijeti stjecaju vještina za radno mjesto nakon školovanja. Kreativnost je važna kada se računalno razmišljanje treba primijeniti na problem. Programiranje je vještina temeljena na kreativnosti, bilo da se koristi za izradu aplikacije, izradu algoritma za pretraživanje ili primjerice dizajn web stranice. Računalno razmišljanje je važno na radnom mjestu jer se sve tvrtke danas bave rješavanjem problema, bilo da se radi o malim poboljšanjima učinkovitosti poslovanja, do stvaranja raznih proizvoda i usluga za potrošače. Bilo koja funkcija suvremenog poslovanja zahtjeva računalno razmišljanje. Ono je postalo ključno na bilo kojem radnom mjestu danas u 21.stoljeću, gdje se toliko toga temelji na podacima. Čak i ako osoba nije programer, nego je na nekom drugom radnom mjestu, bitno je razumjeti i razmišljati o poslovnim problemima pomoću koncepata računalnog razmišljanja koji su prethodno navedeni u ovom radu (dekompozicija, apstrakcija, prepoznavanje uzoraka, algoritam). Sposobnost računalnog razmišljanja korisno je za karijeru u gotovo svakom sektoru, uključujući poslovna i financijska tržišta, turizam, energetiku, zdravstvo, obrazovanje i sl. Radna mjesta trebaju zaposlenike koji su sposobni preuzeti aktivnu ulogu u problemima razmišljanja i stvaranja rješenja[5].

3.Računalne igre u edukaciji

Većina djece danas odrasta u digitalnom društvu i digitalnom dobu. Budući da su tijekom odrastanja navikli na digitalnu tehnologiju, djeca su znatno promijenila svoje načine razmišljanja i obrađivanja informacija od svojih roditelja. Većini djece su računalne igre postale glavni i sastavni dio života, pa je igranje računalnih igara najčešća aktivnost u dječjem slobodnom vremenu. Odgojitelji, odnosno učitelji vjeruju da djeca najbolje uče kroz zabavu, što je i prirodan način za učenje djeteta. Kroz igrice, djeca stječu vještine, a da toga nisu ni svjesna. Stoga ima smisla reći da igrica ima važan doprinos razvoju djeteta, što bi se trebalo uvrstiti i u školski program. Računalna igrica, koja je inače temeljena na igranju i zabavi, može biti orijentirana i na obrazovanje, koja je poznata kao engl.*edutainment.*Takva igrica istovremeno podrazumijeva zabavu i obrazovanje, te igranje ovakve igrice pretpostavlja aktivnost učenja kroz igru[2]. Pojam koji se veže uz oblik poučavanja temeljen na didaktičkim igrama se zove *Game Based Learning* (GBL). Didaktičke igre su igre koje potiču interakciju kod djeteta, te ih nečemu poučavaju i obučavaju. Didaktičke igre navode na razmišljanje, bude maštu i kreativnost, pojačavaju koncentraciju, razmišljanje i logičko zaključivanje[18]. Game Based Learning (GBL) označava učenje temeljeno na igri kroz koje se razvijaju vještine razmišljanja i na učinkovitiji način se uči jer su računalne igre zabavne i privlačne, što doprinosi boljoj izvedbi od tradicionalnih nastavnih planova i metodologija[6].

Igrice su obično implementirane tako da sadrže bogate vizualne elemente kako bi se privuklo djecu u svijet fantazije, te kako bi djeca bila motivirana na zabavan i izazovan način. Razlozi zašto se računalne igre koriste kod učenja su: današnji učenici se radikalno mijenjaju, te ove učenike je potrebno motivirati na nove načine. Osim ovih razloga postoji još nekoliko drugih razloga zašto se računalne igre mogu koristiti kao alat za učenje, primjerice računalne igre imaju pravila, ciljeve, interakciju, sadržaj, priču. Igrice općenito motiviraju djecu te ih poučavaju i potiču na način razmišljanja za rješavanje nekog kompleksnog problema. Potiču ih i na razmišljanje o odlukama koje donose i na to kako će odluke utjecati na druge situacije. Dakle, jedno je sigurno - velika je važnost korištenja igrica u edukaciji.

3.1.Usvojene vještine primjenom računalnih igara u edukaciji

Primjenom računalnih igara u edukaciji, odnosno primjenom strategija temeljenih na GBL, razvijaju se vještine programiranja ali i vještine općenito.

3.1.1.Općenite vještine

Brojne su prednosti koje donosi učenje temeljeno na igri. Učenje temeljeno na igri omogućuje praksu i vježbu. To je najbolji način učenja, posebice jer razvija vještine upravljanja. U igricama se pojavljuju rizične situacije koje jačaju djetetov osobni razvoj, što će se kasnije odraziti i na profesionalni razvoj. Takve vještine nisu naučene u knjigama i teoriji, već samo kroz praksu. Učenje temeljeno na igri potiče i strateško razmišljanje i odlučivanje. Većina računalnih igara zahtijeva od učenika da brzo razmišljaju i koriste logiku kako bi riješili problem. To razvija sposobnost odlučivanja kod učenika i također pomaže razvijanju vještina upravljanja. Vještine kao što su strateško razmišljanje, stavljanje u položaj drugih, razvoj analitičkih i komunikativnih vještina i slično su faktori koji su prisutni u mnogim računalnim igrama za razvijanje osobnih vještina. Osim toga, učenje temeljeno na računalnim igrama poboljšava memoriju. Didaktičke igre i druge računalne igre često imaju određenu strukturu koja ima elemente ponavljanja kroz koju učenici nesvjesno stječu vještine pamćenja i poboljšavaju memoriju, stoga učenje temeljeno na igri zasigurno poboljšava pamćenje kroz praktično učenje. Osim toga, treba spomenuti i prilagodbu na tehnologije. Živimo u svijetu u kojemu dominira tehnologija i gotovo je nemoguće zamisliti radno mjesto gdje računalo nije glavni alat. Učenje temeljeno na igrama omogućuju i prilagodbu na tehnologiju. Korištenje tipkovnice i miša na računalu ili prsta na tabletima i mobilnim uređajima razvija usklađenost između očiju i ruku, što znači da kombinacija fizičke akcije i mentalne pažnje potiče razvijanje memorije, pažnju i koncentraciju[7].

3.1.2.Vještine programiranja

Učenje temeljeno na računalnim igrama razvija i vještine potrebne za programiranje, koje uključuju poznavanje osnovnih koncepata u programiranju. Postoji nekoliko koncepata važnih za programiranje,te su navedeni u nastavku.

Ključni koncept u programiranju je da se određena aktivnost ili zadatak izražava kao slijed (*engl.sequence*) koraka, odnosno uputa koje računalo može izvršiti. Niz uputa za

programiranje određuje ponašanje ili radnju koja bi se trebala izvršiti. Ako je potrebno neku radnju izvršiti određeni broj puta tada postoji mehanizam koji pokreće više puta iste nizove koraka ili uputa, a taj mehanizam se zove petlja (*engl.loops*). Dakle, petlja ponavlja određene korake željenim brojem iteracija. Nadalje, često se javlja situacija da neka radnja uzrokuje drugu radnju i to nazivamo događajima (*engl.events*). Primjerice, gumb za pokretanje koji pokreće videozapis, ili bilo koji drugi događaj koji će proizvesti neku akciju.
Većina današnjih programskih jezika podržava paralelizam (*engl.parallelism*), odnosno nizove koraka koje se istovremeno događaju. Primjer u računalnim igrama je recimo kontinuirano se reproducira neki zvučni zapis, igrač u igri nešto govori uz to, ima mogućnost penjanja, skakanja, ubijanja neprijatelja i slično. Dakle, puno je koraka koji se istovremeno događaju. Još jedan ključni koncept su uvjeti (engl.*conditionals*), sposobnost donošenja odluka na temelju određenih uvjeta, pri čemu je moguće i više ishoda. Ako u računalnoj igri, primjerice, igrač dotakne neprijatelja, gubi život, a inače (ako ga ne dotakne) nastavlja igru. To je primjer petlje. Još je važno spomenuti i podatke. Podaci se moraju negdje pohraniti, dohvatiti i ažurirati. Te podatke je moguće pohraniti pomoću varijabli (*engl.variables*)[8]. Primjer za varijablu je kada se u nekoj računalnoj igri na neki način skupljaju bodovi, a kako se povećava broj bodova tako se povećava vrijednost varijable za bodove, kojom se prikazuje rezultat.

Razumijevanje svih prethodno navedenih koncepata je važno za razvijanje vještina programiranja, a učenici upravo to razvijaju kroz brojne računalne igre.

4.Klasifikacija igara

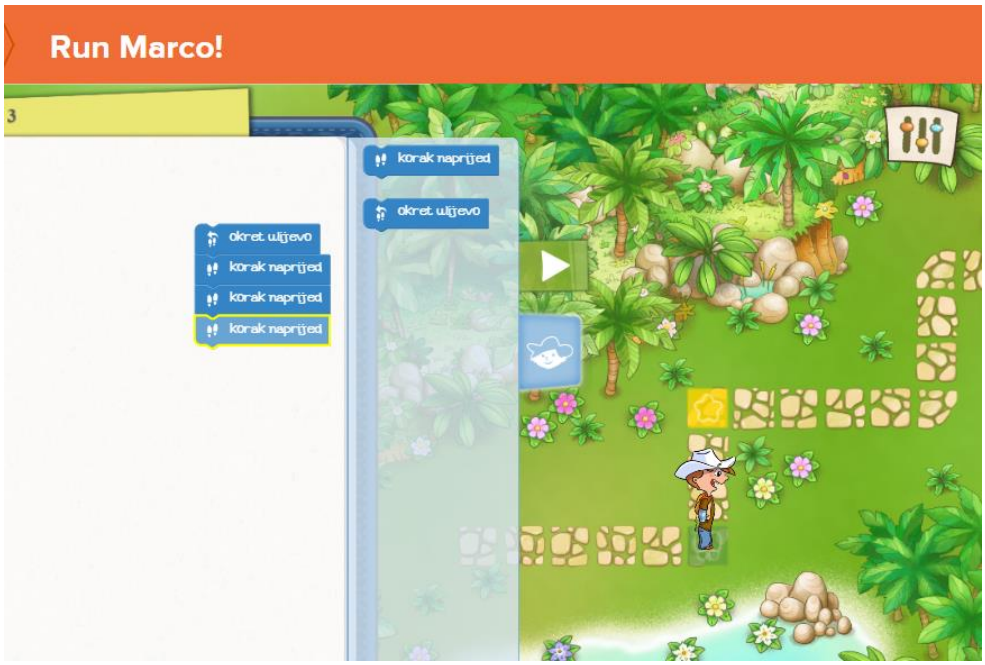
U ovom poglavlju su klasificirane računalne igre te logički i kreativni zadaci, na način da se odnose na usvajanje osnovnih koncepata u programiranju te na usvajanje koncepata računalnog razmišljanja. Neke igre sadrže više mogućnosti i kroz njih se usvaja više od jednog koncepta. U tom slučaju, igra je spomenuta više puta u ovom radu uz sve koncepte na koje se odnosi.

4.1.Usvajanje koncepata programiranja

4.1.1.Usvajanje slijedova koraka

Run Marco

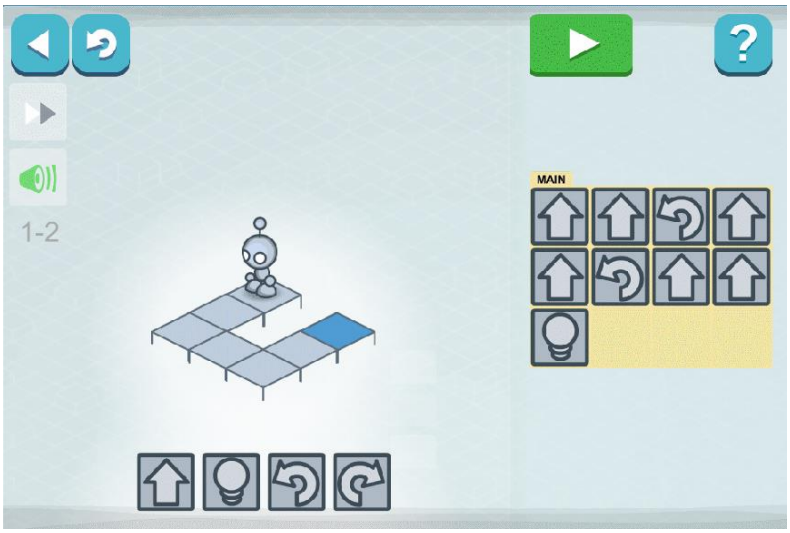
Računalna igra *Run Marco*[22] je osmišljena za učenje programiranja i namijenjena je djeci od šest do dvanaest godina[9]. Djeca kroz ovu igru rade svoje prve korake u programiranju gdje vizualnim naredbama pomažu liku Marku u igri pri rješavanju postavljenih zadataka i prikupljanju dragulja. Usvajaju koncept slijedova koraka, odnosno slijedova naredbi koje se izvršavaju koristeći jednostavne naredbe kao što su korak naprijed, okret u lijevo, okret u desno i sl.
Slika 3 prikazuje kako izgleda igra *Run Marco.* Vidimo da lik Marko u igri treba doći do žutog kvadratića, ali nije okrenut u tom smjeru, što znači da se lik Marko mora prvo okrenuti ulijevo i zatim napraviti tri koraka naprijed, te su takve naredbe i dodane. Na taj način kroz ovu igru učenici usvajaju i vježbaju slijedove koraka.



Slika 3. Prikaz usvajanja slijedova koraka kroz igru Run Marco

Lightbot

Ovu zanimljivu igru[23] je osmislio student Danny Yaroslavski iz Kanade. Predviđena je za učenike koji imaju minimalno 9 godina[12]. Igranjem Lightbot igre učenik usvaja koncept slijedova koraka, odnosno slijedova naredbi koje se moraju izvršiti kako bi se došlo do cilja u igri. Cilj igre je da robot osvijetli sve plave pločice na 3D rešetki. To igrač može učiniti programiranjem svog robota zadajući mu niz uputa. Igrač ima ikonice koje predstavljaju naredbe i mora ih povući u main prozor s desne strane na način da robot dođe do plave ploče. Sljedeća slika prikazuje primjer vježbanja i usvajanja slijedova koraka.



Slika 4. Prikaz usvajanja slijedova koraka kroz igru Lightbot

Vidimo da igrač mora napraviti niz naredbi sljedećim redoslijedom: dva koraka unaprijed, okret u lijevo, dva koraka naprijed, okret u lijevo, dva koraka u naprijed i naredba za osvijetljenje ploče (žaruljica). Svaki sljedeći primjer je sve teži i teži i uvode se neka nova kretanja, primjerice skok u vis i slično, stoga je Lightbot odličan primjer kako kod učenika razviti razumijevanje slijedova koraka, odnosno slijedova naredbi.

Kodable

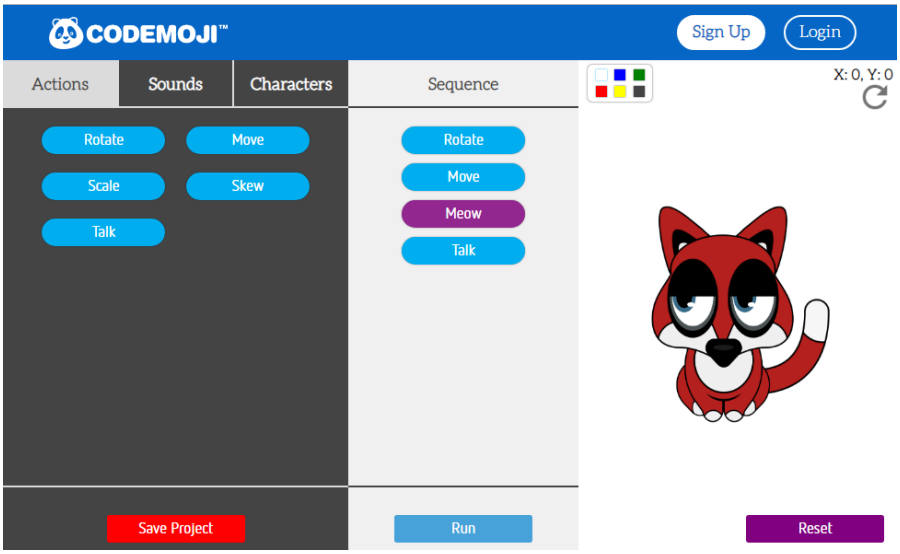
Kodable[24] je igra kroz koju se učenike može naučiti brojne programske koncepte. Izbjegava potpuno korištenje teksta, što ovu igru čini još atraktivnijom za učenike mlađe dobi. Predviđena je za djecu od pet do osam godina. U igri je cilj da se pokupe sve zvjezdice, a da bi se to napravilo potrebno je zadati smjer kretanja strelicama povlačeći ih iz desnog žutog kvadrata lijevo u zadane kvadratiće. Smjer kretanja može biti gore, dolje, lijevo i desno. Kada igrač napravi odgovarajući redoslijed kretanja, pokrene ju na gumb za pokretanje i lik igrača se kreće točno onim smjerovima koji su mu i zadani i na taj način kupi zvjezdice. Pri tome učenici razvijaju razumijevanje slijedova koraka, odnosno slijedova naredbi. Sljedeća slika prikazuje primjer jedne razine u igri prilikom koje igrač usvaja koncept slijedova koraka.



Slika 5. Prikaz usvajanja slijedova koraka kroz igru Kodable

Codemoji

Codemoji igra[25] je igra u kojoj učenici izrađuju vlastite animacije. Sučelje za izradu animacija je vrlo jednostavno i lako za korištenje. Vidimo na sljedećoj slici da učenik sa lijeve strane može odabrati lika u animaciji (kartica Characters). Osim toga, može dodavati zvukove (kartica Sounds) i određene akcije (kartica Actions). Animaciju je jednostavno kreirati povlačenjem željenih naredbi u polje Sequence. Sve naredbe koje učenik povlači su proizvoljne, dakle on sam kreira animaciju kakvu on želi. A sve te naredbe se izvršavaju kao slijed naredbi, odnosno slijed koraka čije je izvršavanje potrebno da bi se dobio dojam animacije. Na taj način učenik usvaja koncept slijedova koraka.



Slika 6. Prikaz usvajanja slijedova koraka kroz igru Codemoji

4.1.2.Usvajanje varijabli

World of variables (Svet Spremenljivk)

Igra[26] se odvija na jednom planetu u svemiru, gdje stanovnici tog planeta imaju problema s uređenjem planeta i trebaju pomoć u isporuci neke robe na druge planete. Roba koja se prevozi u igri ima slična svojstva kao varijable u programiranju. Na taj način učenici mogu naučiti pojam varijable na intuitivan i motivirajući način kroz ovu obrazovnu računalnu igru. Cilj igre je da učenik shvati varijable i dobije osnovnu ideju o tome što se događa u pozadini računala kad se implementiraju varijable[10].

Igra se sastoji od četiri dijela[11]:

- Čišćenje nereda – u ovom dijelu je cilj naučiti učenika deklaraciju varijabli, odnosno rezervaciju memorije. To se provodi na način da igrač mora odabrati odgovarajući spremnik (deklaracija), dodati oznaku (naziv) te dodati sadržaj (dodjela vrijednosti varijabli). Kada se nered pospremi u odgovarajući ambalažu i doda mu se naljepnica vidimo na sljedećoj slici da se na ljubičastu policu automatski slaže pospremljeni nered.



Slika 7. Prvi dio igre Svet Spremenljivk - čišćenje nereda

- Transporteri – cilj ovog dijela je potaknuti učenikovo razumijevanje više tipova varijabli. U igri postoje transporteri koji mogu nositi samo određene proizvode i to je

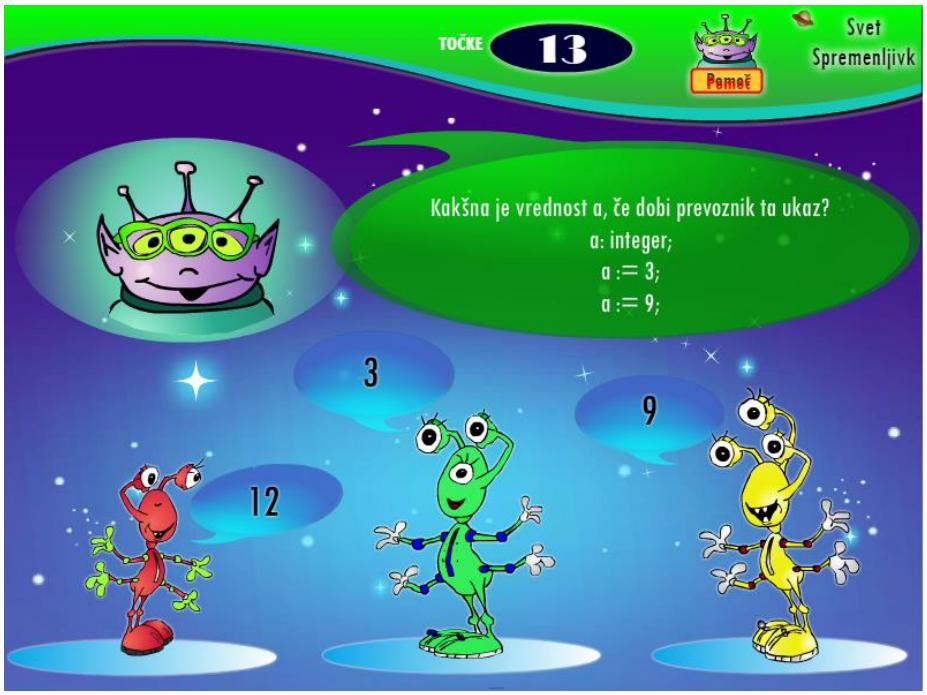
slično različitim vrstama varijabli. Aktivnost se provodi tako što igrač stavlja robu na odgovarajući svemirski brod. Imena planeta kojima se se proizvodi isporučuju imaju ista imena kao neke vrste varijabli. Planeti kojima se isporučuju proizvodi su Integer, Booelan, Real i Char. Sljedeća slika prikazuje kako to i izgleda, dakle na dnu imamo različite vrijednosti koje učenik treba povući na odgovorajući svemirski brod.



Slika 8. Drugi dio igre Svet Spremenljivk – transporteri

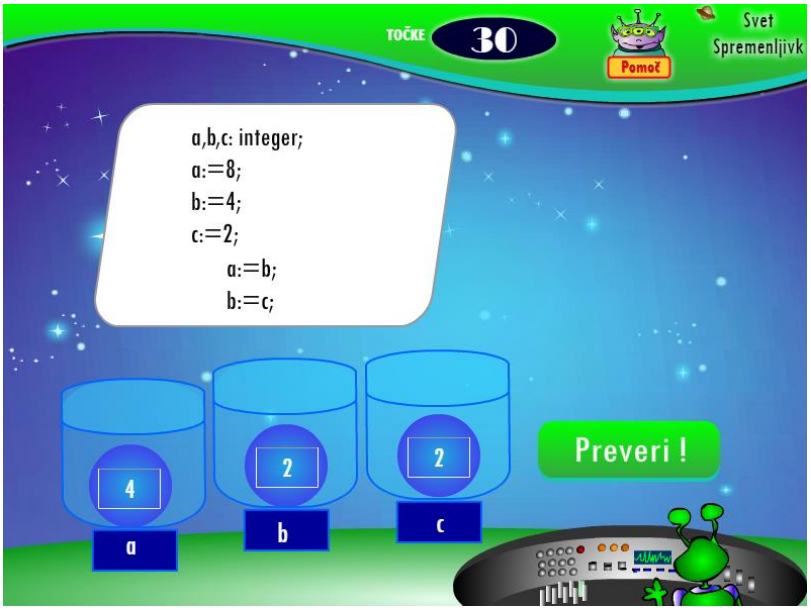
Na taj način je omogućeno razvijanje razmišljanja kod učenika o različitim vrstama varijabli i razumijevanje različitog načina pospremanja različitog tipa podataka, ovisno o tome je li neki podatak cijeli broj, decimalni broj, znakovni niz ili boolean tip.

- Stjecanje vozačke dozvole – cilj ovog dijela igre je učenje učenika u pojmu inicijalizacije, odnosno pohranjivanja vrijednosti u varijablu (npr a:=5). U igri prijevoznici moraju položiti ispit da bi postali dobri prijevoznici. Ispitivač u igri postavlja pitanja o inicijalizaciji varijabli. Igrač mora birati jedan od mogućih odgovora koje ponude tri lika. Primjer pitanja vidimo na sljedećoj slici, očigledno je da je odgovor trećeg lika točan.



Slika 9. Treći dio igre Svet Spremenljivk - stjecanje vozačke dozvole

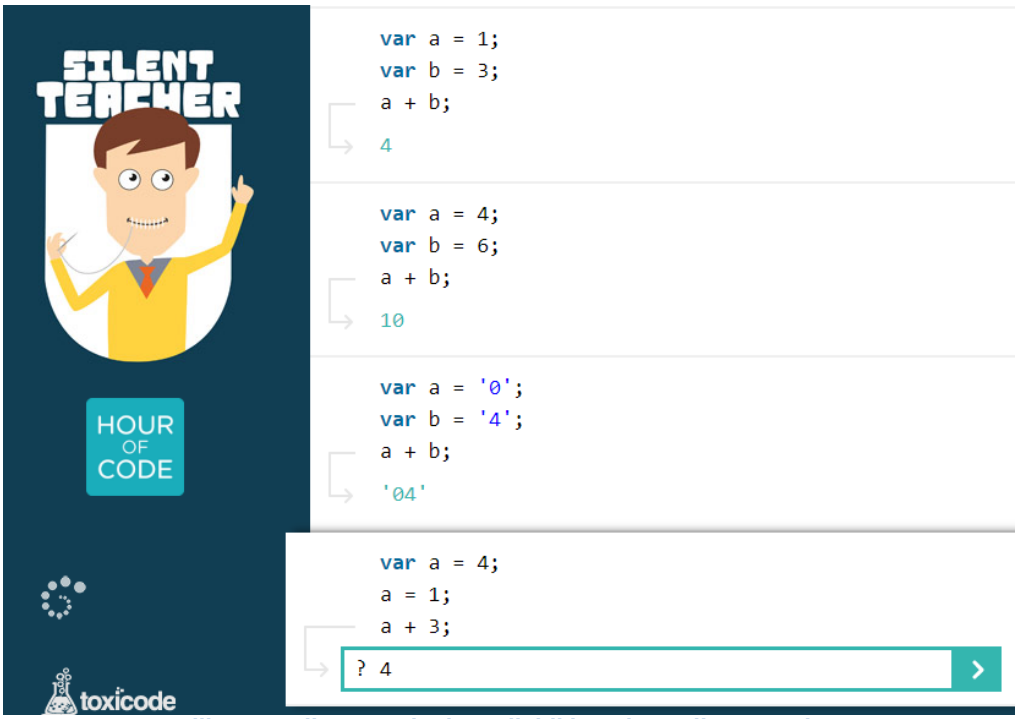
- Pohrana robe – u ovom dijelu igre je cilj učenika naučiti o implicitnom dodjeljivanju vrijednosti varijabli (npr a:=b). Prijevoznik u igri je isporučio neku robu a igrač mu mora pomoći da pohrani robu. To je implementirano na način da igrač mora za svaku varijablu napisati koja je njena vrijednost odnosno pohraniti varijablu u odgovarajući kontejner. Primjer zadatka je vidljiv na sljedcećoj slici, a upisano rješenje vidimo u spremnicima a, b i c.



Slika 10. Četvrti dio igre Svet Spremenljivk

Silent Teacher

Ova zanimljiva igra[27] idealan je primjer kroz koju se uče varijable. Igra se igra bez ikakvih uputa, stoga je igrač primoran sam razumjeti što treba i učiti iz pogreški. Osim što uči varijable, ovu igru možemo povezati i sa drugim predmetom u nastavi, a to je matematika, jer učenik usvaja i razumijevanje matematičkih jednadžbi. Igra je predviđena za učenike od 6 godina pa nadalje, jer je potrebno poznavanje osnova aritmetike za igranje ove igre[16].

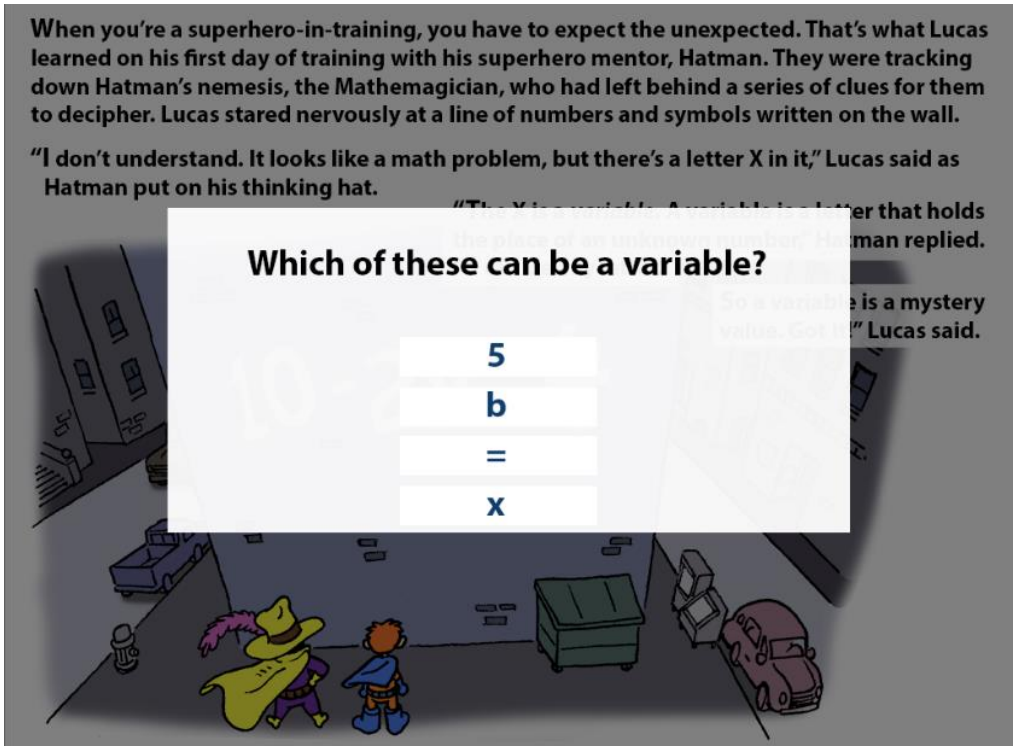


Slika 11. Prikaz usvajanja varijabli kroz igru Silent Teacher

Vidimo na slici da se bez uputa igraču prvo pojavljuju lakša pitanja koja izgledaju kao jednostavna aritmetika, a nakon toga se postupno uvode teža pitanja. Budući da za vrijeme igranja igre nema uputa, 'učitelj' šuti, to je prednost jer je igra tako namijenjena svim igračima diljem svijeta bez obzira na njihov jezik. Ono što ovu igru čini zanimljivom je to da kada je odgovor igrača neispravan, prikazat će mu se drugo pitanje iste vrste. Ako igrač napravi više pogrešaka na istom tipu pitanja, 'učitelj' će mu postavljati slična pitanja nekoliko puta, čak i ako učenik pravilno odgovori na pitanje, samo kako bi se uvjerio da je stvarno razumio koncept tog tipa pitanja. Kada igrač pravilno odgovara na pitanja pojavljuje se smiješak, a ako odgovori krivo pojavljuje se tužan emotikon.

Hatman

U ovoj igri[28], superheroj Lucas uči sve o varijablama, izrazima i jednadžbama kako bi pomogao Hatmanu u svojoj potrazi za neprijateljem, Mathemagicianom, ali oni to ne mogu učiniti sami. U ovoj akcijskoj, matematičkoj priči, djeca moraju odgovoriti na pitanja s višestrukim odabirom kako bi tijek priče išao dalje. U igri je učenicima sve objašnjeno na najjednostavniji mogući način, tako da nove stvari uče kroz zanimljivu akcijsku priču, a onda im se postavlja pitanje iz tog dijela igre. Na slici vidimo prikaz primjer pitanja koji se postavlja učenicima. Radi se o varijablama. Učenik bi trebao znati točno odgovoriti na pitanje, a to je da *x* i *b* mogu biti varijable. Ovu igru možemo povezati i sa drugim nastavnim predmetom, matematikom, jer se uče jednadžbe.

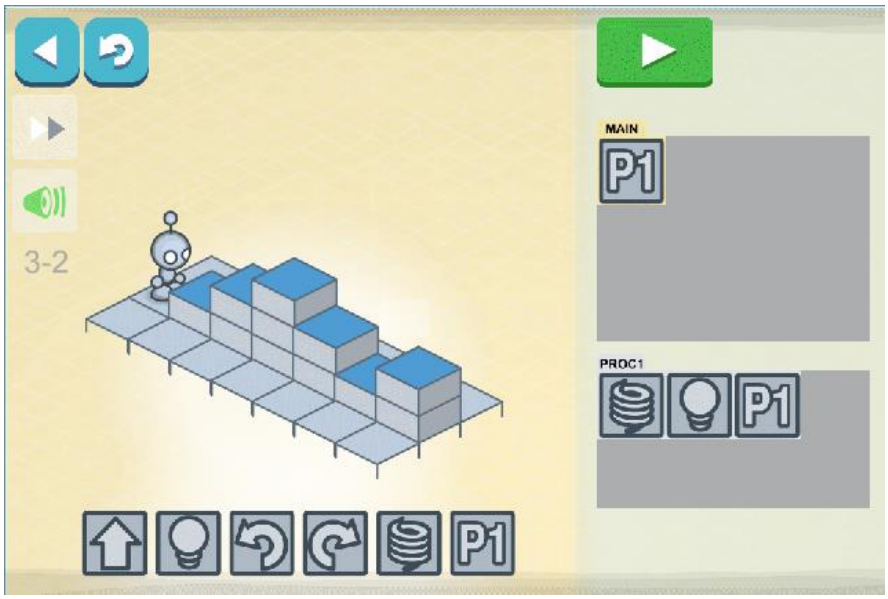


Slika 12. Prikaz usvajanja varijabli kroz igru Hatman

4.1.3.Usvajanje petlji

LightBot

Ova igra[23], predviđena za učenike koji imaju minimalno devet godina, je već prethodno spomenuta u radu kao primjer za usvajanje koncepta nizova, međutim kroz neke igre se usvaja više kocepata, a ova igra je odličan primjer igre kroz koju se zadaci u igri odnose na usvajanje više koncepata programiranja. Cilj igre je da igrač dovede robota na plave kvadratiće i osvijetli ih. To može učiniti pomoću odgovarajućih naredbi koje jednostavno povlači u prozore s desne strane. Sljedeća slika prikazuje primjer kroz koji vidimo kako učenik može razumjeti princip petlji u programiranju.

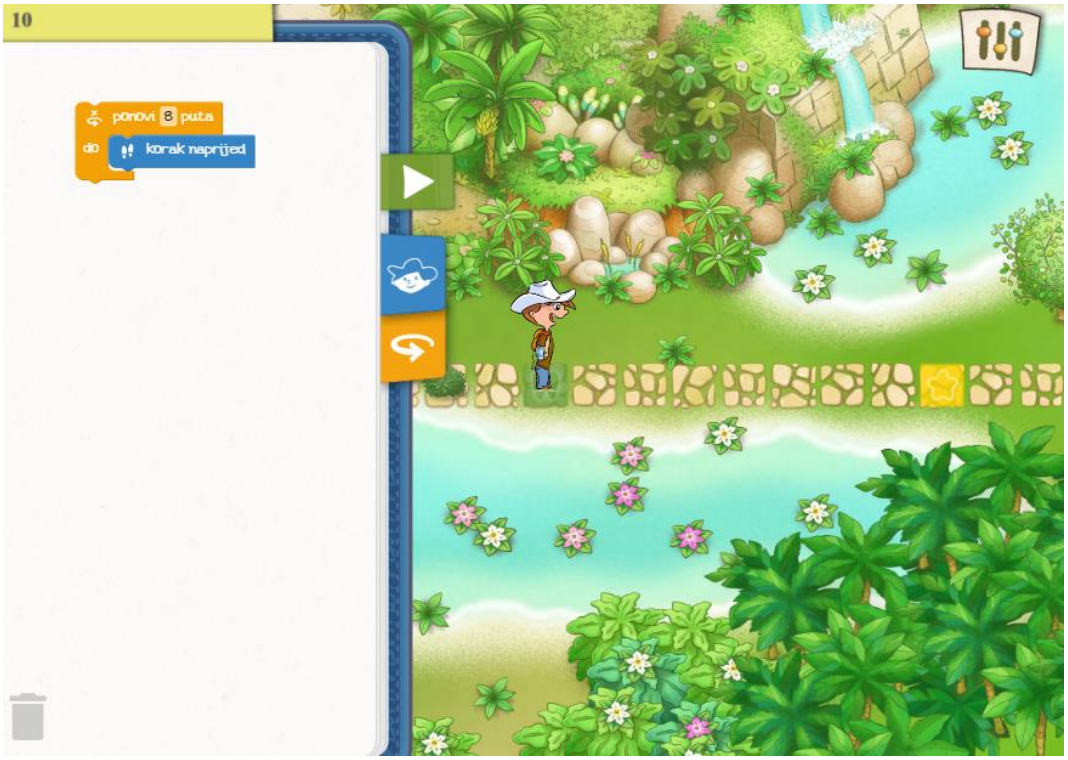


Slika 13. Prikaz usvajanja petlji kroz igru LightBot

Pogledamo li sliku vidimo da robot mora napraviti skok i upaliti svjetlo na kvadratiću, zatim opet mora napraviti skok i upaliti svjetlo na kvadratiću, te se taj identičan postupak ponavlja sve do zadnjeg plavog kvadratića. Učenik kroz ovu računalnu igru razvija svijest o tome da se ponekad javlja situacija kada je iste korake potrebno ponavljati puno puta. Isto tako, kroz ovakvu igru može naučiti da postoje načini kako pojednostaviti ovakav tip problema, odnosno da postoji način kako će se automatski ponavljati određeni koraci u igrici više puta, koristeći petlje. Vidimo na slici da se u PROC1 polje dodaju naredbe za skok i paljenje svjetla, te naredba P1 koja označava da je to petlja, a u MAIN prozoru se „pozove“ izvršavanje te petlje.

Run Marco

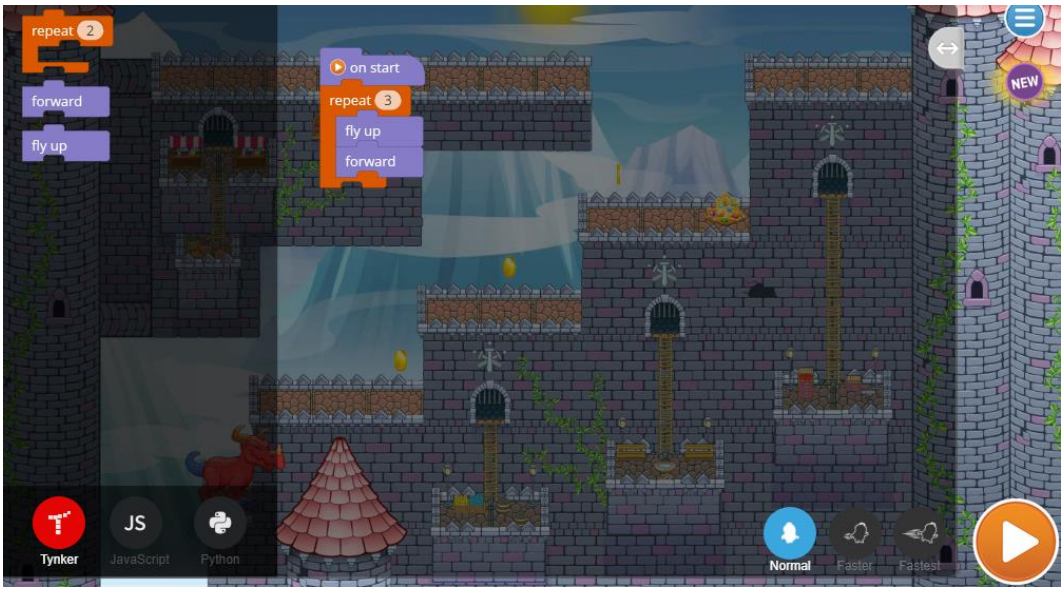
Ova igra[22] je već prethodno spomenuta kao igra kroz koju se usvaja koncept slijedova koraka. Osim usvajanja tog koncepta, izvrsna je i za usvajanje koncepta petlji. Cilj igre je da lik Marco dođe do žutog kvadratića, a do tog kvadratića ima 8 koraka unaprijed. Koristit će naredbu koja se zove 'korak naprijed' i naredba se mora staviti unutar bloka 'ponovi x puta', odnosno unutar petlje. Broj koji treba upisati je 8, jer toliko koraka Marco mora napraviti do žutog kvadratića. Na ovaj način, učenici će vrlo jednostavno shvatiti princip petlji te će razumjeti kada se javlja potreba za ponavljanjem nekog koraka više puta.



Slika 14. Prikaz usvajanja petlji kroz igru Run Marco

Dragon Dash

Ova zanimljiva igra predviđena je za djecu od 5 do 8 godina[15]. Može se pronaći na Tynker-u[29]. Nakon dizajnirana svog vlastitog lika-zmaja po želji, učenik mora upotrijebiti naredbe kako bi vodio zmaja kroz zabavne prepreke u potrazi za blagom i novčićima. Učenik mora naredbama kretati zmaja kroz složene dvorce te pronalaziti načine kako skupiti sve novčiće. Sljedeća slika prikazuje primjer jedne razine iz igre Dragon Dash, koja je izvrstan primjer za usvajanje koncepta petlji kroz ovu igru.

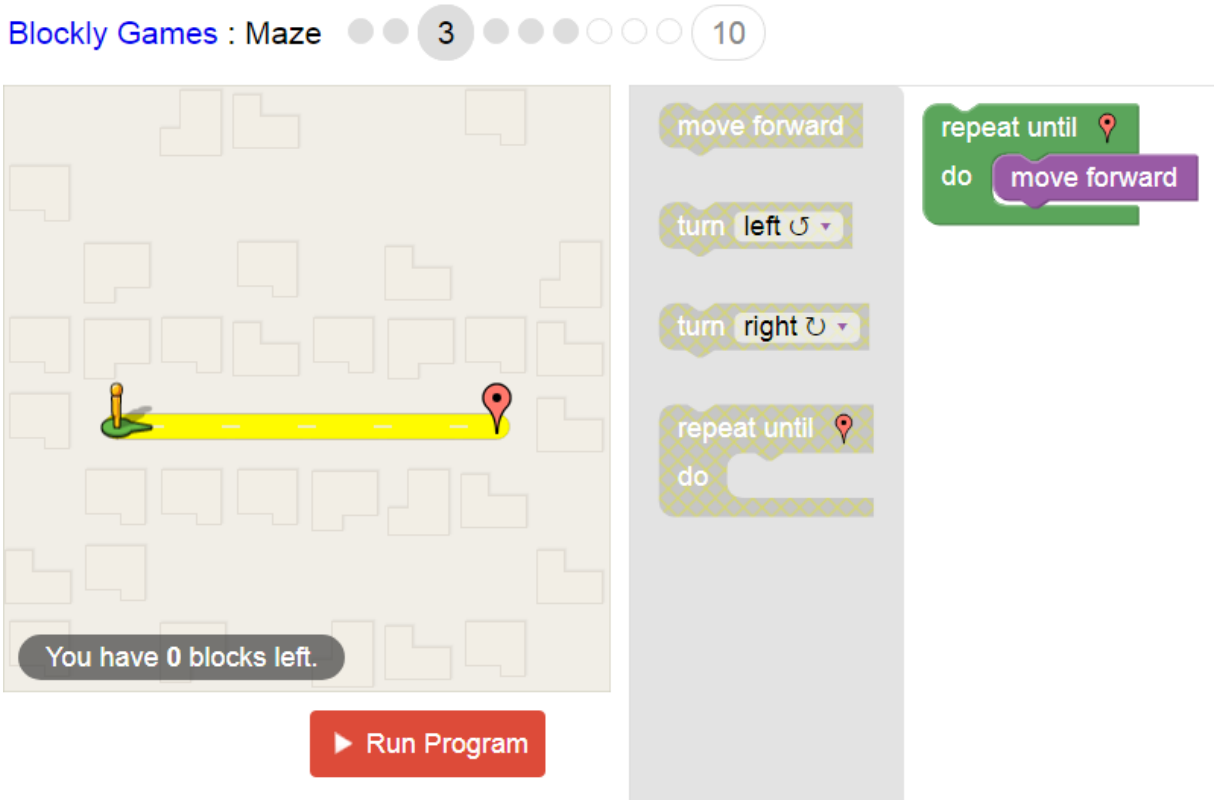


Slika 15. Prikaz usvajanja petlji kroz igru Dragon Dash

Učenik prilikom ove igre može zaključiti da zmaj prvo mora skočiti pa hodati ravno da bi skupio novčič, a ako malo bolje pogleda dvorac vidjet će da istu stvar mora napraviti 3 puta. Tu dolazi do zaključivanja da se neki koraci ponavljaju određeni broj puta i učenik počinje razumijevati koncept petlji i zašto se koriste petlje.

Blockly

Ova igra[30] predstavlja primjer igre u kojoj učenici mogu usvajati koncept petlji u programiranju. Učenici moraju riješiti zadatak povlačenjem slagalica i na taj način rješavaju zadatak. Vidimo u primjeru da igrač mora doći do određene lokacije odnosno mora napraviti nekoliko koraka unaprijed i tu učenik treba shvatiti princip petlji, odnosno da se one koriste kada se neka naredba izvršava više puta. Mora još shvatiti da postoji neki uvjet koji označava kraj izvršavanja petlji, odnosno petlja će stati sa izvršavanjem kada uvjet bude ispunjen. U ovom slučaju uvjet je da igrač dođe do crvene lokacije. U ovoj igri kroz lagane primjere učenik razvija shvaćanje o petljama.

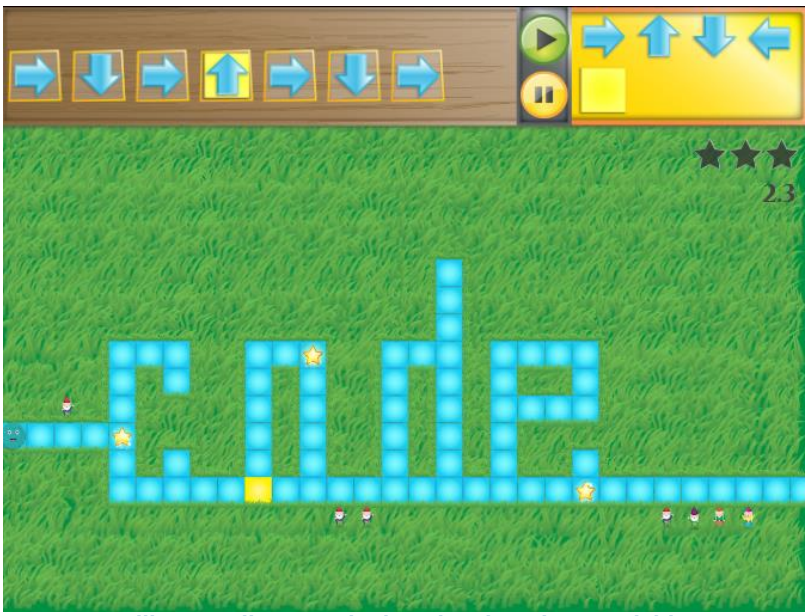


Slika 16. Prikaz usvajanja petlji kroz igru Blockly

4.1.4.Usvajanje uvjeta

Kodable

Igranjem ove računalne igre[24] i najmlađi (od pet do osam godina) započinju svoje početke u svijetu programiranja. Kroz ovu igru učenici mogu naučiti što su uvjeti i kada se javlja potreba za uvjetima unutar nekog problema. Igrač skuplja zvjezdice zadajući niz naredbi pomoću strelica koje predstavljaju smjer kretanja. Sljedeća slika prikazuje primjer iz igre Kodable kroz koju se može vidjeti kako je implementirano usvajanje uvjeta kod učenika.



Slika 17.Prikaz usvajanja uvjeta kroz igru Kodable

Vidimo na slici da igrač mora promijeniti smjer kretanja u lijevo kad dođe do žutog kvadratića, da bi pokupio zvjezdicu. Tu se javlja uvjet koji bi glasio: Ako je igrač došao do žutog kvadratića, onda idi lijevo. Učenici jednostavno mogu riješiti ovaj tip problema tako što će u nizu strelica povući žuti kvadratić preko odgovarajuće strelice (četvrta naredba na slici).

RoboGarden

RoboGarden[31] je zanimljiva, lako razumljiva obrazovna igra kroz koju učenici uče osnovne koncepte. Na slici je prikazan primjer učenja uvjeta kroz ovu igru. Vidimo na slici da robot skuplja satelite. Prije skupljanja on mora provjeriti boju ćelije. Ako je ćelija zelene boje, to je super i može slobodno ići naprijed i skupiti idući satelit. A ako je crvene, onda je put blokiran kamenjem i robot ne bi smio ići preko kamenja.

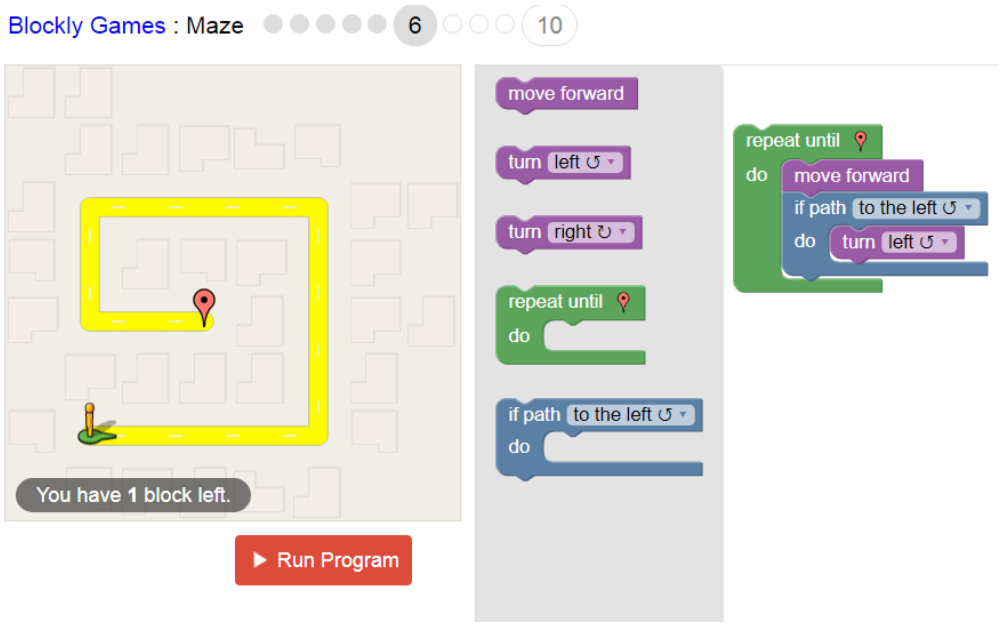


Slika 18. Prikaz usvajanja uvjeta kroz igru RoboGarden

Vidimo na primjeru da robot prvo ide dva koraka unaprijed, zatim se postavlja uvjet u kojem se provjerava je li boja ćelije jednaka zelenoj boji. Ako jest, može ići dalje, dva koraka unaprijed i pokupiti idući satelit.

Blockly

Ova igra[30] predstavlja primjer igre u kojoj učenici mogu usvajati koncept uvjeta u programiranju. Učenici moraju riješiti zadatak povlačenjem slagalica kako bi stvorili niz naredbi koje bi riješile određeni problem. Vidimo u primjeru da igrač mora doći do cilja pomičući se naprijed, a u slučaju da dođe do skretanja, mora se zarotirati ulijevo, i tu učenik mora znati primjeniti uvjet. Ova igra je korisna jer se uče i još neki drugi koncepti kao slijedovi naredbi i petlje, te je već spomenuta i objašnjena prethodno u tekstu.



Slika 19. Prikaz usvajanja uvjeta kroz igru Blockly

4.2.Usvajanje koncepata računalnog razmišljanja

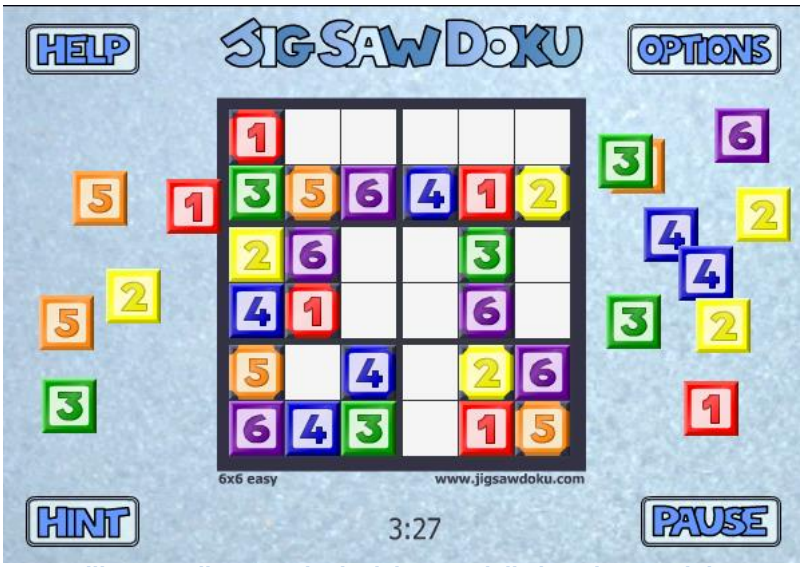
Koncepti računalnog razmišljanja su prethodno već objašnjeni u ovom radu, to su dakle dekompozicija, prepoznavanje uzoraka, apstrakcija i algoritmi.

4.2.1.Usvajanje dekompozicije

Dekompozicija se odnosi na razbijanje složenog problema u manje probleme. Složen problem je onaj kojeg na prvi pogled ne znamo kako lako riješiti.

Sudoku Jigsawdoku

Učenici bi kroz igranje ove igre[32] trebali znati razbiti problem u manje dijelove, a rješavanjem sudokua potiče se uočavanje točno određenog pravila raspoređivanja sličica, odnosno brojeva kao što vidimo na slici. Važno je razumjeti pravilo i znati ga primijeniti u različitim situacijama.

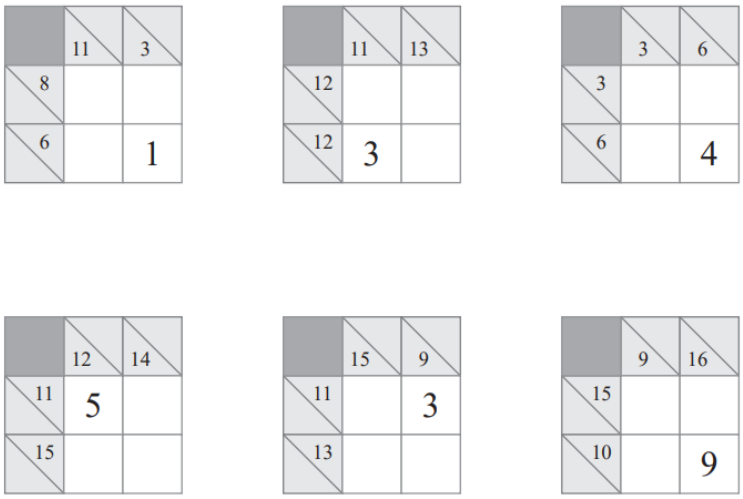


Slika 20. Prikaz usvajanja dekompozicije kroz igru Sudoku

Kakuro for Kids

U ovoj igri[33] učenik mora popuniti prazna polja koristeći znamenke od 1 do 9, i to tako da suma vodoravnog bloka polja bude jednaka broju s njihove lijeve strane, te suma vertikalnog bloka polja bude jednaka broju na vrhu. Pri tome se svaki broj može upotrijebiti samo jednom po bloku. Na slici je prikazan primjer takvog tipa zadataka 3x3 gdje je ponuđen već jedan broj koji olakašava rješavanje zadataka, no postoje i teže verzije koje zahtijevaju malo više

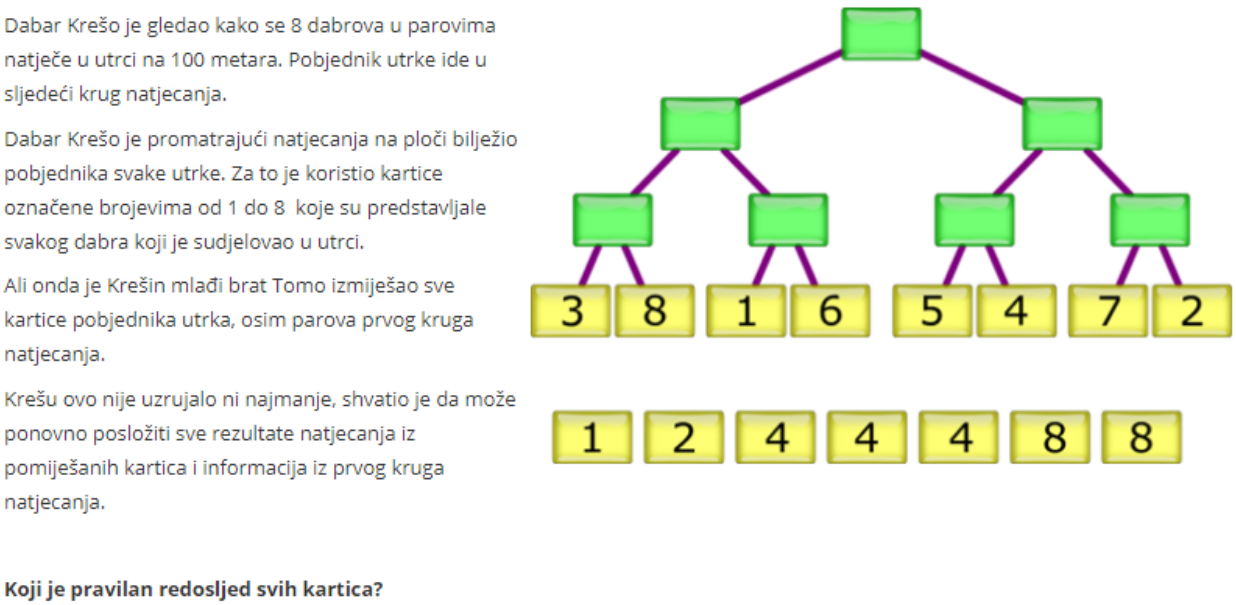
razmišljanja. Primjerice zadaci u kojima nije ponuđen niti jedan broj ili primjerice zadaci koji su 4x4, 5x5 i slično.



Slika 21. Prikaz usvajanja dekompozicije kroz igru Kakuro

Kakuro igra je tip složenog problema kojeg učenik mora znati rastaviti na manje probleme i rješavati korak po korak, odnosno broj po broj.

Natjecanje dabrova



Slika 22. Prikaz usvajanja dekompozicije kroz zadatak Natjecanje dabrova

Natjecanje dabrova predstavlja priču u kojoj je lik Krešo gledao utrke i zabilježio je pobjednike svake faze na ploči. Trkači su nosili iste brojeve, od 1 do 8, tijekom cijelog turnira. Svaki trkač je predstavljen jednom brojčanom kartom. Nakon završetka turnira, mlađi

brat Tomo je izmiješao sve karte osim onih iz prve faze turnira. To su karte predstavljene žutim kvadratićima. Zadatak učenika je da sve brojčane karte postavi na ispravnu poziciju, odnosno brojeve unutar zelenog kvadratića mora rasporediti po ploči. Na slici vidimo prikazano rješenje ovoga zadatka. Da bi ga učenik ispravno riješio ovaj složeni, logički zadatak, mora rješavati dio po dio, odnosno „granu po granu“ u ovom stablu. Na taj način primjenjuje tehniku dekompozicije. Zadatak je namijenjen učenicima od 8 do 10 godina i može se pronaći na Dabru, KiloDabar (5.i 6.razred OŠ)[20].

4.2.2.Usvajanje prepoznavanja uzoraka

Nakon dekompozicije, odnosno razbijanja problema na manje probleme, svaki od tih problema se može pogledati pojedinačno i prepoznavanje uzoraka podrazumijeva traženje sličnosti između manjih problema.

Thinking Myself

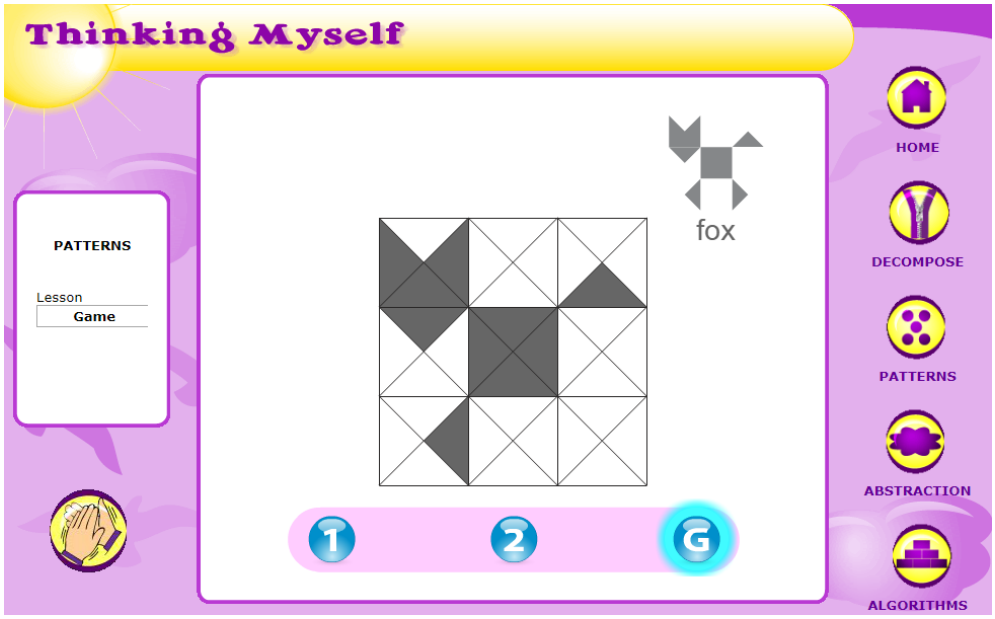
Ovo je primjer igre[34] u kojoj se vježbaju sve tehnike računalnog razmišljanja. Vidimo na idućoj slici da s desne strane imamo sve lekcije koje možemo odabrati. Na prvoj slici je pokazan primjer lekcije koja se odnosi na uzorke.



Slika 23. Prikaz usvajanja prepoznavanja uzoraka kroz igru Thinking Myself – prvi dio

Učenik bi trebao odabrati koji je uzorak sljedeći u zadanom nizu. To je naravno zeleni kvadratić. Ukoliko učenik dobro riješi ovaj zadatak, pretpostavlja se da razumiije što se misli pod pojmom uzorci, pa mu se postavlja idući zadatak s uzorcima koji je malo drukčiji. Taj zadatak je prikazan na idućoj slici. Učeniku je zadan neki lik a on bi trebao znati prepoznati

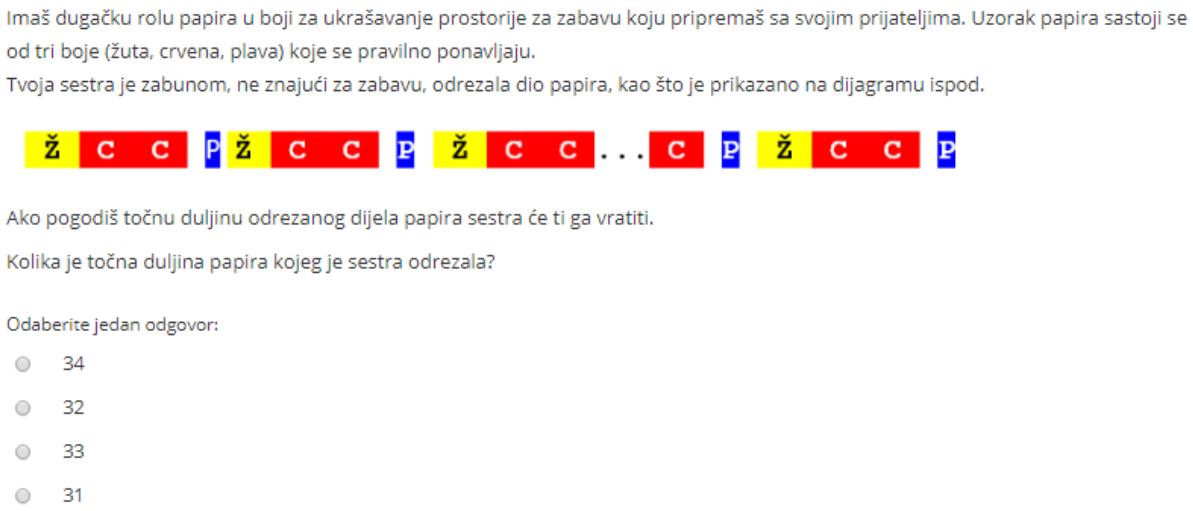
uzorke i nacrtati isti taj lik. Kroz ovakav tip zadatka učenik usvaja prepoznavanje uzoraka kao tehniku računalnog razmišljanja.



Slika 24. Prikaz usvajanja prepoznavanja uzoraka kroz igru Thinking Myself – drugi dio

Zastava za proslavu

U ovom zadatku iz Dabra[21] je zadan niz slova koja su obojena sa tri različite boje (žuta, crvena i plava). Niz je zadan u određenom poretku koji se ponavlja. U sredini niza je odsječen dio slova i zadatak učenika je da pogodi veličinu niza koji je odsječen. Da bi učenik riješio zadatak prvo mora pogledati niz i uočiti uzorak brojeva koji se ponavlja. Na taj način učenik traži sličnosti u nizu brojeva i usvaja prepoznavanje uzoraka. Uzorak koji se ponavlja u ovom zadatku je ŽCCP i rješenje zadatka je 31.



Slika 25. Usvajanje prepoznavanja uzoraka kroz igru Zastava za proslavu

4.2.3.Usvajanje apstrakcije

Apstrakcija podrazumijeva fokusiranje samo na važne informacije, ignorirajući nevažne detalje.

Thinking Myself

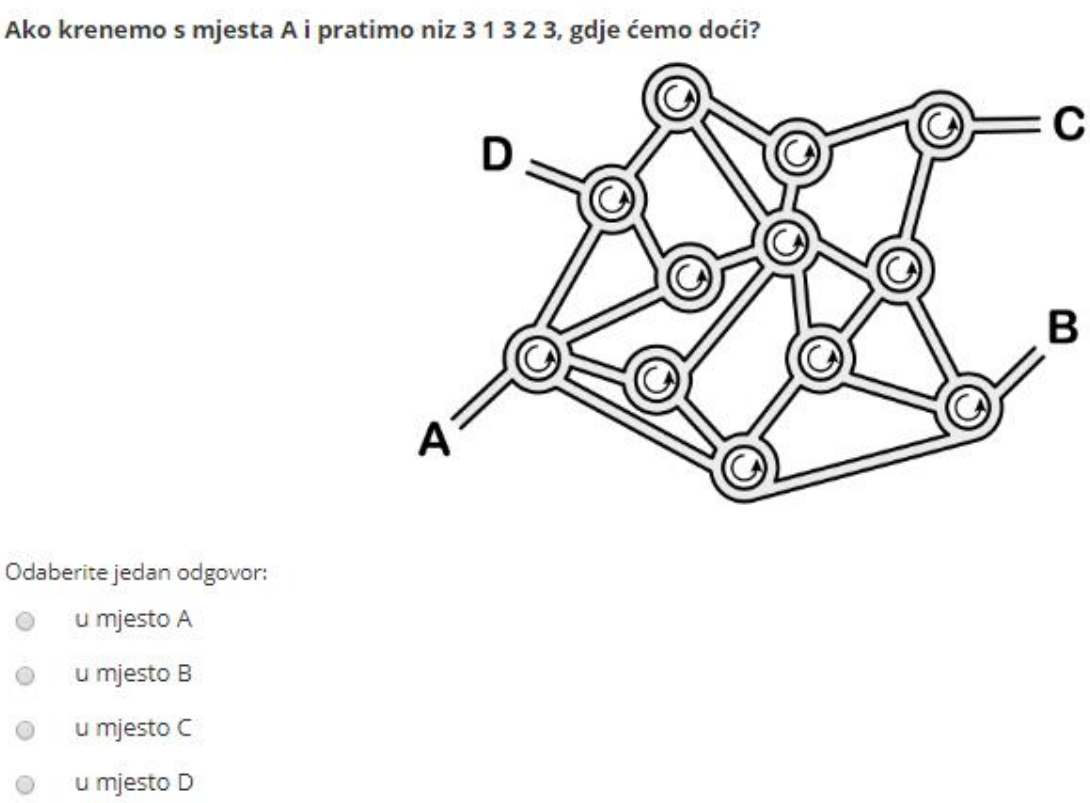
Apstrakcija se fokusira samo na važne informacije, pri tome zanemaruje manje bitne detalje, tako da se može pronaći rješenje koje bi odgovaralo mnogim različitim stvarima. Sljedeća slika prikazuje primjer igre[34] kroz koju se usvaja apstrakcija. Vidimo na slici primjer apstrakcije a to je rečenica: ____&____makes____. To je recimo baza koja je uvijek ista, a prazna polja se mogu popuniti sa puno različitih mogućnosti. Vidimo da vrijednost prva dva prazna polja ovisi o trećem polju. Primjerice, traže se dvije boje koje zajedno daju ljubičastu,, a to su plava i crvena. Zadatak učenika je da ispravno posloži zadane sličice. Na taj način učenik usvaja apstrakciju kao tehniku računalnog razmišljanja.



Slika 26. Prikaz usvajanja apstrakcije kroz igru Thinking Myself

Grad kružnih tokova

Ovaj primjer zadatkaje primjer sa Dabra[20]. U ovom zadatku je zadana slika kružnih tokova i niz brojeva 31323 koji predstavlja redoslijed izlaza iz kružnih tokova kako bi vozač trebao izlaziti. Zadatak učenika je odrediti gdje će vozač na kraju doći. Na taj način se učenik fokusira samo na ono bitno, na izlaze iz kružnih tokova, a pri tome zanemaruje ono ostalo, nevažno.Na taj način usvaja apstrakciju. Rješenje je B.

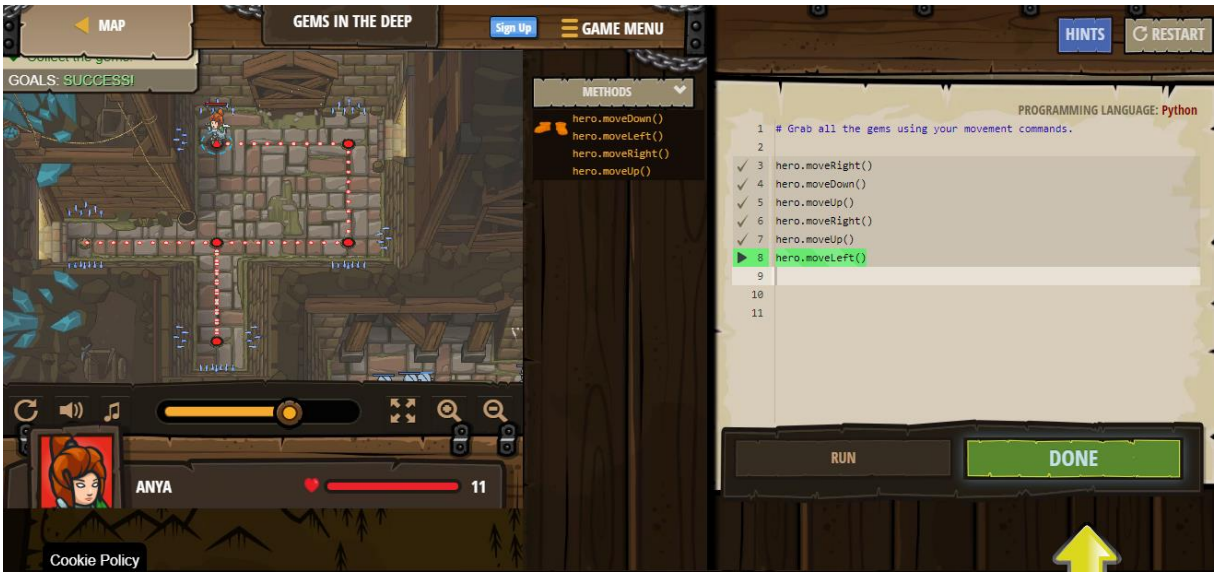


Slika 27. Prikaz usvajanja apstrakcije kroz igru Grad kružnih tokova

4.2.4.Usvajanje algoritama

Code Combat

Ova zanimljiva igra[35] je predviđena za malo starije učenike, od 13 godina i starije[12], iako ju mogu igrati i mlađi ukoliko im netko da smjernice. Ova igra je specifična po tome što osim što razvija algoritamsko razmišljanje, istovremeno i podučava sintaksu programskog jezika. Prije početka igranja je moguće odabrati programski jezik Python ili JavaScript. Cilj igre je da lik igrača pokupi zlatnike a naredbama učenik treba odrediti smjer kretanja lika. U primjeru kojeg prikazuje iduća slika, prikazan je programski jezik Python. Vidimo na slici da su u sredini već ponuđene četiri metode, a na učeniku je samo da iskoristi ponuđeno i natipka kod u desnom bijelom prozoru.



Slika 28. Prikaz usvajanja algoritamskog razmišljanja kroz igru Code Combat

Ponuđene metode su hero.moveDown(), hero.moveUp(), hero.moveLeft() i hero.moveRight(). Poziv tih metoda pokreće lika da se kreće u zadanom smjeru. Iako učenik ovdje ne mora sam tipkati kod jer mu je već ponuđeno ono što mu treba, učenik ipak uči osnove sintakse koja će mu možda jedan dan trebati. U ovom primjeru polako usvaja da se primjerice objektima pristupa sa točkom ili da metode/funkcije pišemo za zagradom i slično. Zanimljivo je kod ove igre da nema jedinstvenog rješenja. Ono što je bitno je da zlatnici budu pokupljeni, a kojim putem će to točno biti, nije važno. Dakle učenik sam mora osmisliti smjerove kretanja i redoslijed kupljenja zlatnika. Na taj način učenik razvija algoritamsko razmišljanje, odnosno sposobnost odabira postupaka za rješavanje nekog problema.

Code Monkey

Code Monkey je obrazovna igra[36] koja je osvojila brojne nagrade. Pomaže učenicima usvojiti algoritamski način razmišljanja na vrlo jednostavan način. Na svakoj razini u igrici učenik treba koristiti različite kodove kako bi majmun stigao do odredišta, odnosno do banane. Ovdje učenik mora koristiti svoju kreativnost kako bi dovršio zadatak. Vrlo je jednostavna igra i pogodna za početnike i učenike nižih razreda, bez velikih komplikacija[14]. Sljedeća slika prikazuje primjer jedne razine u igri. Na slici vidimo da su na dnu već ponuđene naredbe koje se trebaju koristiti u igri, tako da sve što učenik mora napraviti je kliknuti na odgovarajuću naredbu, ne mora sam tipkati kod. Naredbama mora dovesti majmuna do banane, dakle potrebno ga je zarotirati u lijevo te napraviti određeni broj koraka.

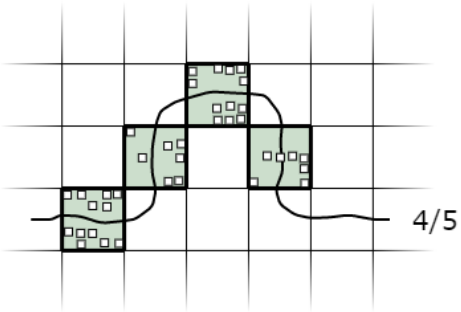
Potreban broj koraka je jednostavno vidjeti tako što samo povuče ravnalo iz gornjeg lijevog kuta i prikaže se točna udaljenost između banane i majmuna. U ovom slučaju potreban broj koraka je 12. Igra nije komplicirana i iziskuje samo malo logičkog razmišljanja. Učenik kroz ovu igru, logički razmišljajući, pronalazi način kako riješiti problem u igri i na taj način razvija sposobnost algoritamskog načina razmišljanja.



Slika 29. Prikaz usvajanja algoritamskog razmišljanja kroz igru Code Monkey

Initial Conditions

U ovoj igri[37] učenik se suočava s mrežom s rijekama i mora staviti određeni broj gradova na rijeku, tako da rijeka prolazi kroz njih i tako da ne postoje dva grada koja se dodiruju, niti vodoravno niti okomito. Ovaj problem je tipičan problem koji se javlja u području umjetne inteligencije za probleme pretraživanja[14]. To naravno nije objašnjeno učeniku, on samo vidi igru s problemom postavljanja gradova, ali igranje ove igre će ga izazvati na algoritamsko razmišljanje.

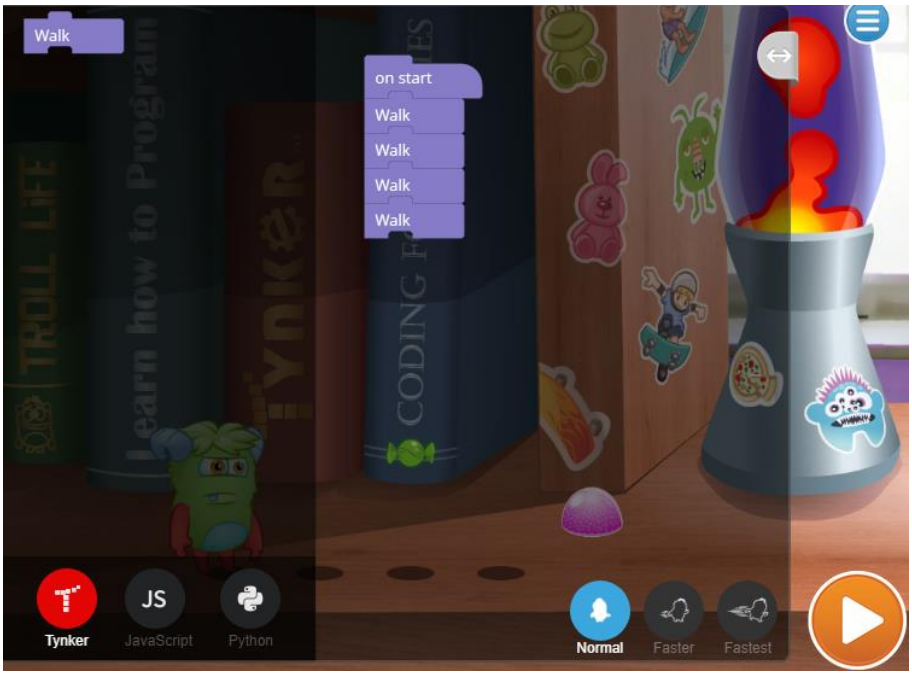


Place 5 towns on the river. Two towns cannot touch, horizontally
or vertically. Left click to place towns.

Slika 30. Prikaz usvajanja algoritamskog razmišljanja kroz igru Initial Conditions

Candy Quest

Ova igra se može pronaći na Tynker-u[38]. Sljedeća slika prikazuje primjer te igrice. Potrebno je glavnog lika dovesti do ljubičastog predmeta. Učenik ima na raspolaganju naredbu walk koja predstavlja jedan korak i mora sam zaključiti koliko je puta potrebno primjeniti tu naredbu da bi doveo lika do željenog predmeta. U ovom slučaju to su 4 koraka. Na taj način razvija razmišljanje o rješavanju nekog problema, te razmišljanje o slijedu naredbi koje se moraju izvršiti.



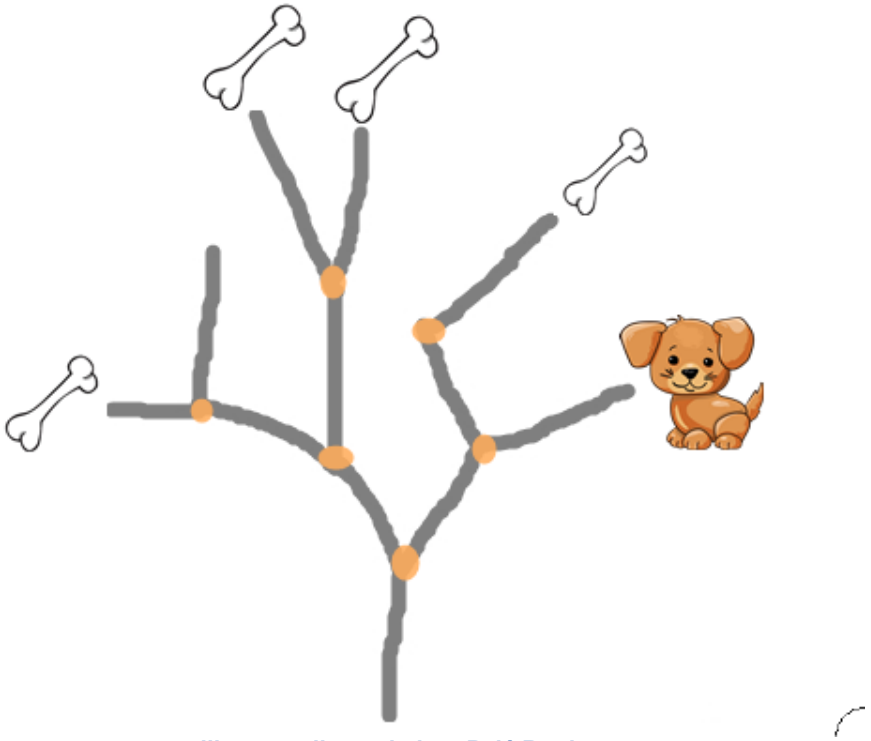
Slika 31. Prikaz usvajanja algoritamskog razmišljanja kroz igru Candy Quest

5.Primjeri zadataka za učenike

U nastavku slijede osmišljeni primjeri koji bi mogli doprinijeti usvajanju koncepata računalnog razmišljanja i programiranja.

Psić Boni

U igri se nalazi psić Boni na početku jedne ceste. Ceste su podijeljenje po žutim krugovima. Psić Boni mora pojesti sve četiri kosti koje se nalaze na slici. Potrebno je pronaći najmanji broj koraka koje psić Boni mora napraviti kako bi pojeo sve kosti. Minimalni broj koraka je 15. Ova igra doprinosi usvajanju koncepata računalnog razmišljanja. Primjerice, dekompozicija se može primijeniti za ovaj zadatak, raščlanjivanjem zadatka na dijelove. Prvi dio se odnosi na broj koraka potrebnih za prvu kost, drugi za drugu i tako dalje. Na kraju se zbroje svi koraci za sve 4 kosti. Osim tehnike računalnog razmišljanja, tu se još i razvijaju općenite vještine kao primjerice sposobnost logičkog zaključivanja.

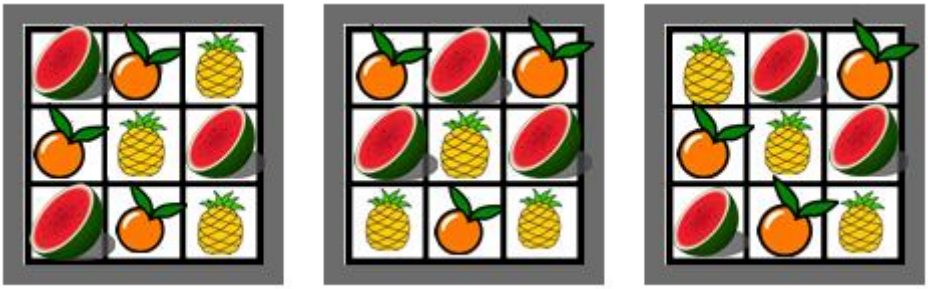


Slika 32. Prikaz primjera Psić Boni

Sudoku

Drugi osmišljeni primjer je igra sudoku. Rješavanjem sudokua potiče se uočavanje točno određenog pravila raspoređivanja sličica. Važno je pravilo razumjeti ga i primijeniti u različitim situacijama. Kroz ovu igru se razvija koncept računalnog razmišljanja dekompozicija. Osim toga, Sudoku jača i druge sposobnosti kao što su logika i memorija, te sposobnost donošenja odluke. Kroz ovu igru učenika se uči rješavanju problema, povećanju koncentracije i traži od učenika da usmjeri pažnju na zadatak, te jača kreativnost.

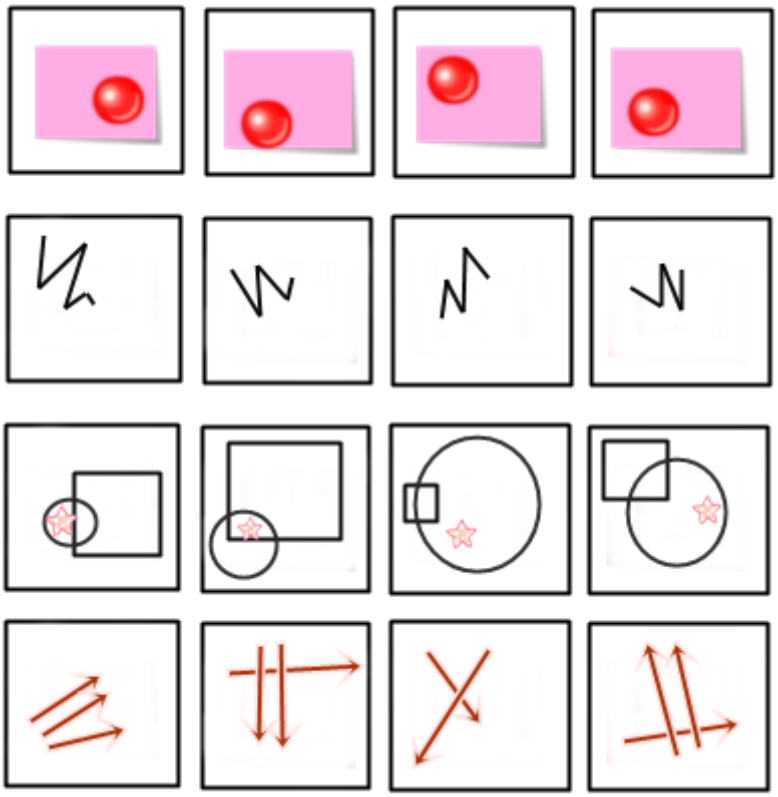
U ovom primjeru se traži od učenika da od prikazana tri kvadratića odabere koji je ispravan, uzevši u obzir pravilo da sve sličice moraju biti različite u svkom retku i svakom stupcu. Točan odgovor je treći kvadrat. Ovo je samo primjer Sudoku-a gdje učenik bira točan odgovor među ponuđenim rješenjima. A kako bi razvijao sve sposobnosti koje Sudoku nudi, preporuča se rješavanje zadataka bez ponuđenih odgovora, posebice većih dimenzija kao što su 4x4, 5x5 i slično.



Slika 33. Prikaz primjera Sudoku

Izbaci uljeza

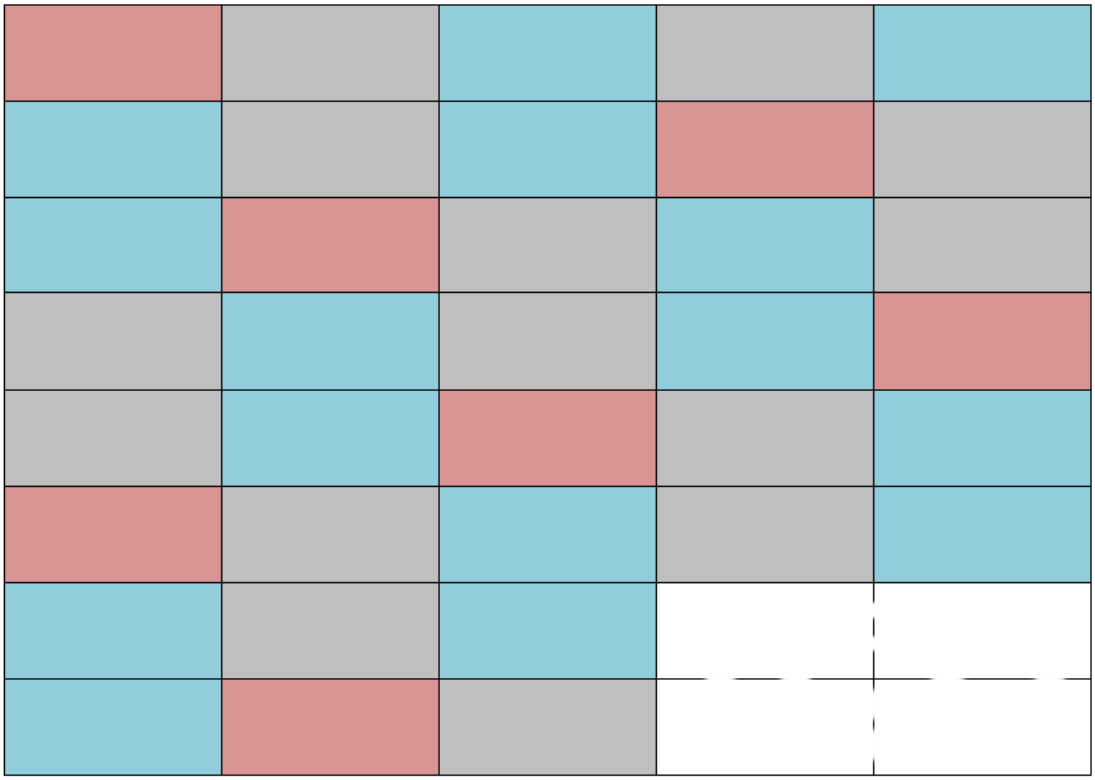
Treći primjer se odnosi na zadatke u kojima je potrebno izbaciti uljeza. Kroz ovakav tip zadataka, koji je omiljeni među djecom, učenik razvija pažnju te pamćenje. Od koncepata računalnog razmišljanja, učenik razvija tehnike apstrakcije, na način da se fokusira na sličicima na ono što je bitno, jer uljez je samo jedan i to onaj koji se po nekoj sitnici razlikuje od ostalih sličica. Fokusirajući se samo na tu sitnicu učenik primjenjuje tehniku apstrakcije, važnu tehniku računalnog razmišljanja. Na slici je prikazan primjer četiri osmišljena zadataka. U prvom redu uljez je druga sličica jer jedino na drugoj sličici crvena loptica dodiruje rub lika, a na svim ostalim sličicama loptica se nalazi unutar lika ne dodirujući niti jedan rub. U drugom redu uljez je prva sličica jer je lik na prvoj sličici nacrtan pomoću pet crti, dok su svi ostali nacrtani pomoću četiri crte. U trećem redu uljez je druga sličica jer se cvjetić nalazi unutar kvadrata jedino na toj slici, a na ostalim slikama cvjetić je izvan kvadrata. U četvrtom redu uljez je treća slika jer sadrži dvije strelice, dok ostale sadrže po tri strelice.



Slika 34. Prikaz primjera Izbaci uljeza

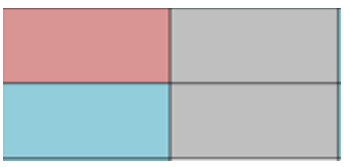
Nastavi niz

Ovaj primjer zadatka se odnosi na nastavljanje niza. U zadatku su četiri praznine koje treba popuniti odgovarajućim bojama. Ovom igrom učenik razvija tehniku uočavanja uzoraka kao tehniku računalog razmišljanja. On mora uočiti uzorak, odnosno poredak boja koji se ponavlja kako bi znao riješiti ovaj zadatak. To je način rješavanja problema na temelju prethodnog rješenja. Pogleda li se poredak boja odozgo prema dolje uočava se sljedeći uzorak – ružičasta, plava, plava, siva, siva.



Slika 35. Prikaz primjera Nastavi niz

Rješenje zadatka je vidljivo na idućoj slici.



Slika 36. Prikaz rješenja primjera Nastavi niz

6.Zaključak

Važnost informatike u obrazovanju posljednjih godina prepoznata je u svim školskim sustavima, kao način poboljšanja produktivnijeg rada s računalima kod djece. Programiranje u informatici kod učenika razvija logički način razmišljanja koji pridonosi kreativnom rješavanju problema, te ih na neki način priprema za budućnost razvijajući brojne vještine kao što su upornost, uspješan timski rad, vježbanje načina razmišljanja, pedantnost i sl.

Programiranje, kao važan element informatike, je i zahtjevan element, a često i nedovoljno zanimljiv učenicima. Kako bi se sadržaj informatike učinio zanimljivijim, kod učenika mlađe dobi se uvode didaktičke igre. Na taj način se već u osnovnoj školi razvija računalno razmišljanje. Računalno razmišljanje je proces razmišljanja uključen u formuliranje problema i izražavanje njegovog rješenja na takav način da bude strojno razumljiv i da se može strojno izvršiti. Četiri glavne tehnike računalnog razmišljanja su dekompozicija, apstrakcija, prepoznavanje uzoraka te algoritmi. U diplomskom radu sam napravila klasifikaciju igara koje se odnose na usvajanje koncepata računalnog razmišljanja. Smatram da se kroz svaku igru koju sam pronašla usvaja barem jedan od koncepata računalnog razmišljanja a kroz neke i više od jednoga.

Većini učenika su računalne igre postale glavni i sastavni dio života, pa je igranje računalnih igara najčešća aktivnost u dječjem slobodnom vremenu, a to je prava prilika da se računalne igre iskoriste za edukaciju i učenje. Osim općenitih vještina, kod djeteta se razvijaju i vještine programiranja. U radu je napravljena i klasifikacija igara koje se odnose na usvajanje koncepata programiranja kao što su slijedovi koraka, varijable, petlje i uvjeti. Mislim da svaka od navedenih igara omogućava različite načine razmišljanja te razumijevanje svih prethodno navedenih koncepata je važno za razvijanje vještina programiranja.

Smatram da bi ovakve računalne igre trebalo što više uvoditi u edukaciji u nastavi informatike. No osim informatike, razni logički i kreativni zadaci bi trebali biti uključeni što više i u druge nastavne predmete, prvenstveno u matematiku.

7.Popis literature

[1] Zašto svako dijete treba naučiti programirati, Futura-učionica. Preuzeto 10.7.2018. s http://futura-ucionica.hr/zasto-svako-dijete-treba-nauciti-programirati/

[2] K., Wong, K.W., Fung, C.C. and Khine, M.S., The relationship between game genres, learning techniques and learning styles in educational computer games, 2008.

[3] Computational thinking, Wikipedia. Preuzeto 28.08.2018. s https://en.wikipedia.org/wiki/Computational_thinking

[4] Introduction to computational thinking, BBC. Preuzeto 29.08.2018. s https://www.bbc.com/bitesize/guides/zp92mp3/revision/1

[5] The benefits od computational thinking, British Computer Society. Preuzeto 30.08.2018. s https://www.bcs.org/content/ConWebDoc/55416

[6] What is GBL?, Triseum. Preuzeto 31.08.2018. s https://triseum.com/gbl/

[7] 5 great benefits of game-based learning in soft skills training, Game Learn. Preuzeto 01.09.2018.s https://www.game-learn.com/game-based-learning-in-soft-skills-training-5-great-benefits/

[8] K.Brennan, M.Resnick, New frameworks for studying and assessing the development of computational thinking, AERA 2012

[9] Trči Marko i nauči programiranje, Učitelji. Preuzeto 11.09.2018.s http://ucitelji.hr/trci-marko-i-nauci-programiranje/

[10] Svet Spremenljivk, Hrast. Preuzeto 12.09.2018. s http://hrast.pef.uni-lj.si/games/website/svetSpremenljivk.html

[11] Learning programming with serious games, Academia. Preuzeto 13.09.2018. s http://www.academia.edu/3764990/Learning_programming_with_serious_games

[12] 12 games that teach kids to code – and are even fun,too, Venturebeat. Preuzeto 14.09.2018. s https://venturebeat.com/2014/06/03/12-games-that-teach-kids-to-code/

[13] Top 10 Coding Games, Eduonix. Preuzeto 18.09.2018. s https://blog.eduonix.com/software-development/top-10-coding-games-kids-learn-programming/

[14] Learning Programming through Games and Contests: Overview, Characterisation and Discussion, Ioinformaics. Preuzeto 20.09.2018. s https://ioinformatics.org/journal/v10_2016_39_60.pdf

[15] Dragon Dash, Tynker. Preuzeto 28.09.2018 s https://www.tynker.com/hour-of-code/dragon-dash

[16] About Silent Teacher, Silentteacher. Preuzeto 15.10.2018.s http://silentteacher.toxicode.fr/about
[17] A.Csizmadia, P.Curzon, M.Dorling, S.Humphreys, T.Ng, C.Selby, J.Woollard, Computational thinking, A guide for teachers, Computing At School, 2015
[18] Didaktičke igre – što su i čemu služe?, Pino Toys. Preuzeto 3.12.2018.s https://pino-toys.rs/didakticke-igracke-sta-su-i-cemu-sluze/
[19]Mikro Dabar, Loomen. Preuzeto 5.12.2018. s https://loomen.carnet.hr/mod/quiz/review.php?attempt=875700#q1
[20] KiloDabar, Loomen. Preuzeto 5.12.2018. s https://loomen.carnet.hr/mod/quiz/attempt.php?attempt=875745&page=4
[21] MiliDabar, Loomen. Preuzeto 5.12.2018.s https://loomen.carnet.hr/mod/quiz/review.php?attempt=875819
[22] Run Marco, BrainPop. Preuzeto 6.12.2018.s https://www.brainpop.com/games/runmarco/
[23] Lightbot, Lightbot. Preuzeto 6.12.2018. s http://lightbot.com/flash.html
[24] Kodable, Game Kodable. Preuzeto 6.12.2018. s https://game.kodable.com/play?user=n
[25] Codemoji, Codemoji. Preuzeto 6.12.2018. s https://www.codemoji.com/workshop.php
[26] Svet spremenljivk, Hrast. Preuzeto 6.12.2018 s http://hrast.pef.uni-lj.si/games/website/jar/SvetSpremenljivkIgra/index.html
[27] Silent Teacher, Toxi Code. Preuzeto 6.12.2018. s http://silentteacher.toxicode.fr/
[28] Hatman, Education. Preuzeto 6.12.2018. s https://www.education.com/game/hatman/
[29] Dragon Dash, Tynker. Preuzeto 6.12.2018. s https://www.tynker.com/lesson/v2/host
[30] Blockly Games, Blockly Games. Preuzeto 6.12.2018. s https://blockly-games.appspot.com/maze?lang=en
[31] Robo Garden, Robo Garden. Preuzeto 6.12.2018. s https://robogarden.ca/
[32] Jigsawdoku, Jigsawdoku. Preuzeto 6.12.2018. s http://www.jigsawdoku.com/
[33] Kakuro for Kids, Math in English. Preuzeto 6.12.2018.s https://www.mathinenglish.com/puzzleskakuro.php
[34] Thinking Myself, Thinking Myself. Preuzeto 6.12.2018. s http://games.thinkingmyself.com/
[35] Code Combat, Code Combat. Preuzeto 6.12.2018 s https://codecombat.com/play
[36] Code Monkey, Play Code Monkey. Preuzeto 6.12.2018. s https://www.playcodemonkey.com/
[37] Initial Conditions, Reheated. Preuzeto 6.12.2018. s https://reheated.org/games/initial/
[38] Candy Quest, Tynker. Preuzeto 6.12.2018. s https://www.tynker.com/lesson/

8.Popis slika

Slika 1. Temeljne tehnike računalnog razmišljanja 6
Slika 2. Primjer računalnog razmišljanja za rješavanje složenog problema 9
Slika 3. Prikaz usvajanja slijedova koraka kroz igru Run Marco 16
Slika 4. Prikaz usvajanja slijedova koraka kroz igru Lightbot 16
Slika 5. Prikaz usvajanja slijedova koraka kroz igru Kodable 17
Slika 6. Prikaz usvajanja slijedova koraka kroz igru Codemoji 18
Slika 7. Prvi dio igre Svet Spremenljivk - čišćenje nereda 19
Slika 8. Drugi dio igre Svet Spremenljivk – transporteri 20
Slika 9. Treći dio igre Svet Spremenljivk - stjecanje vozačke dozvole 21
Slika 10. Četvrti dio igre Svet Spremenljivk 21
Slika 11. Prikaz usvajanja varijabli kroz igru Silent Teacher 22
Slika 12. Prikaz usvajanja varijabli kroz igru Hatman 23
Slika 13. Prikaz usvajanja petlji kroz igru LightBot 24
Slika 14. Prikaz usvajanja petlji kroz igru Run Marco 25
Slika 15. Prikaz usvajanja petlji kroz igru Dragon Dash 26
Slika 16. Prikaz usvajanja petlji kroz igru Blockly 27
Slika 17.Prikaz usvajanja uvjeta kroz igru Kodable 28
Slika 18. Prikaz usvajanja uvjeta kroz igru RoboGarden 29
Slika 19. Prikaz usvajanja uvjeta kroz igru Blockly 30
Slika 20. Prikaz usvajanja dekompozicije kroz igru Sudoku 31
Slika 21. Prikaz usvajanja dekompozicije kroz igru Kakuro 32
Slika 22. Prikaz usvajanja dekompozicije kroz zadatak Natjecanje dabrova 32
Slika 23. Prikaz usvajanja prepoznavanja uzoraka kroz igru Thinking Myself – prvi dio 33
Slika 24. Prikaz usvajanja prepoznavanja uzoraka kroz igru Thinking Myself – drugi dio 34
Slika 25. Usvajanje prepoznavanja uzoraka kroz igru Zastava za proslavu 35
Slika 26. Prikaz usvajanja apstrakcije kroz igru Thinking Myself 36
Slika 27. Prikaz usvajanja apstrakcije kroz igru Grad kružnih tokova 37
Slika 28. Prikaz usvajanja algoritamskog razmišljanja kroz igru Code Combat 38
Slika 29. Prikaz usvajanja algoritamskog razmišljanja kroz igru Code Monkey 39
Slika 30. Prikaz usvajanja algoritamskog razmišljanja kroz igru Initial Conditions 40
Slika 31. Prikaz usvajanja algoritamskog razmišljanja kroz igru Candy Quest 41
Slika 32. Prikaz primjera Psić Boni 42

Slika 33. Prikaz primjera Sudoku 43
Slika 34. Prikaz primjera Izbaci uljeza 44
Slika 35. Prikaz primjera Nastavi niz 45
Slika 36. Prikaz rješenja primjera Nastavi niz 45