

Roboti kao alati u obrazovanju

Ilić, Vida

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:688868>

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

Download date / Datum preuzimanja: **2024-07-16**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Vida Ilić

ROBOTI KAO ALATI U OBRAZOVANJU

Diplomski rad

Voditelj rada:
v. pred. dr. sc. Goran Igaly

Zagreb, 2024.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Zahvaljujem svom mentoru na neizmjernom strpljenju, povjerenju i podršci.

Zahvaljujem i svima drugima koji su mi pružili podršku tokom pisanja ovog rada, a posebno Ernestu Wendlingu, svom mentoru iz kolegija Metodička praksa iz informatike u srednjoj školi, koji mi je omogućio da održim sate s robotima u MIOC-u te kolegici Karli Juranović koja mi je dopustila da opišem našu aktivnost u ovom radu.

Rad posvećujem svojoj majci.

Sadržaj

Sadržaj	iv
Uvod	1
1 Roboti	3
1.1 Što čini robota	6
2 Roboti u obrazovanju	11
2.1 Logo i LEGO	11
2.2 Scratch	17
2.3 micro:bit	19
2.4 Arduino	21
2.5 Makeblock	24
2.6 Natjecanja iz robotike	29
2.7 Tablica	32
3 Roboti u nastavi	35
3.1 Kurikulum	35
3.2 Aktivnosti	41
3.3 Roboti u MIOC-u	49
4 Prilozi	51
4.1 Prilog 1	51
4.2 Prilog 2	59
Bibliografija	63

Uvod

Koncept robota prvo je bio prisutan u znanstveno fantastičnim knjigama, predstavama i filmovima, ali ubrzo se pojavio i u stvarnom životu te se razvio i proširio u mnoga polja ljudskog djelovanja. U posljednjih nekoliko desetljeća roboti su sve češće prisutni i u obrazovanju. Roboti su zamjenili i olakšali mnoge ljudske manualne poslove stoga ih je važno razvijati te podučavati djecu o konceptima robotike, automatizacije i programiranja. Glavni cilj ovog rada je dati generalni pregled robotike u obrazovanju i nastavi Informatike s naglaskom na učenje programiranja.

Ovaj rad na početku prikazuje povijest robotike, razne definicije pojma robot te osnovne komponente koje stroj mora imati da bi ga se moglo klasificirati kao robota. Potom je proučena povijest robotike u obrazovanju, nakon čega su prikazani neki od trenutno najrasprostranjenijih robota koji služe kao alati za učenje programiranja i drugih računalnih vještina. Na kraju je prikazana uloga robota u nastavi Informatike uz primjere nekih aktivnosti i zadataka koji se mogu provoditi na nastavi Informatike koristeći robote.

Poglavlje 1

Roboti

Riječ robot prvi put se spominje 1920. u drami R. U. R. češkog autora Karel Čapek, a izvedena je iz češke riječi *robota* koja u prijevodu znači prisilni rad (rad koji odrađuju robovi ili, jednostavno, rob). Koncept robota, doduše, bio je prisutan mnogo ranije. Prvi „roboti”, kao što su na primjer Arhitina drvena leteća ptica (*The Pigeon*) koja se pokretala parom ili vodeni satovi (klepsidre) uz koje su humanoidne figurice na razne načine označavale protok intervala vremena, pojavljuju se stotinama godina prije Krista.

Postavlja se pitanje, koje je uopće značenje riječi robot? Čapek je robota opisao kao „čovjekolikoga stroja sposobnoga za rasuđivanje, a konstruiranoga kako bi zamijenio ljudski rad u tvornicama” (Leksikografski zavod M. Krleža, 2013 – 2024). U današnjem svijetu roboti ne moraju nužno biti ljudskog oblika te njihova korisnost nadilazi sam rad u tvornicama iako su industrijski roboti ipak najrašireniji.



Slika 1.1: Plakat za predstavu R. U. R.



Slika 1.2: Vaucansonov Patak

Automatski uređaji ili automati su uređaji koji „na vanjski podražaj automatski obavljaju određene radnje ili daju složene odgovore, odnosno podatke” (Nikolić, 2019). Smatraju se pretečom robota pa se povijest robota i povijest automata često isprepliću i poistovjećuju. Tako su *The Pigeon* i klepsidre s početka poglavlja ustvari automati. Automatima je često svrha, osim korisnosti i olakšavanja ljudskog rada, bila i zabavljanje čovjeka pa su tako kroz povijest stvoreni brojni automati koji su se kretali i/ili svirali. Također su kroz povijest ljudi često težili čovjekolikim automatima, a ta težnja često je prisutna i danas. Ismail Al-Jazari, arapski izumitelj iz 12. stoljeća koji je proučavao i izrađivao automate, često se naziva ocem robotike. Njegov glazbeni automat (brod sa četiri automatizirana glazbenika koji je plutao po jezeru) namijenjen za uveseljavanje vladara i njegovih gostiju smatra se prvim programabilnim automatom u povijesti jer je postojala mogućnost promjene ritma glazbe. Al-Jazari je napisao knjigu *Automata, Knjiga znanja o genijalnim mehaničkim uređajima* koja se smatra jednim od najvrjednijih dokumenata za projektiranje, proizvodnju i montažu strojeva u povijesti (Nikolić, 2019). Od povijesnih automata vrijedno je spomenuti i mehaničkog viteza Leonarda da Vincija koji je mogao samostalno raditi određene pokrete kao što su ustajanje i sjedanje, pokreti rukama i pomicanje vizira. Osim toga, poznati Patak Jacquesa de Vaucansona značajan je u području modeliranja kretanja živih bića pomoću mehaničkog stroja. Patak je bio sposoban kretati se i raditi stvari kao da je živ, a najzanimljivija je njegova sposobnost za gutanje i probavljanje hrane.



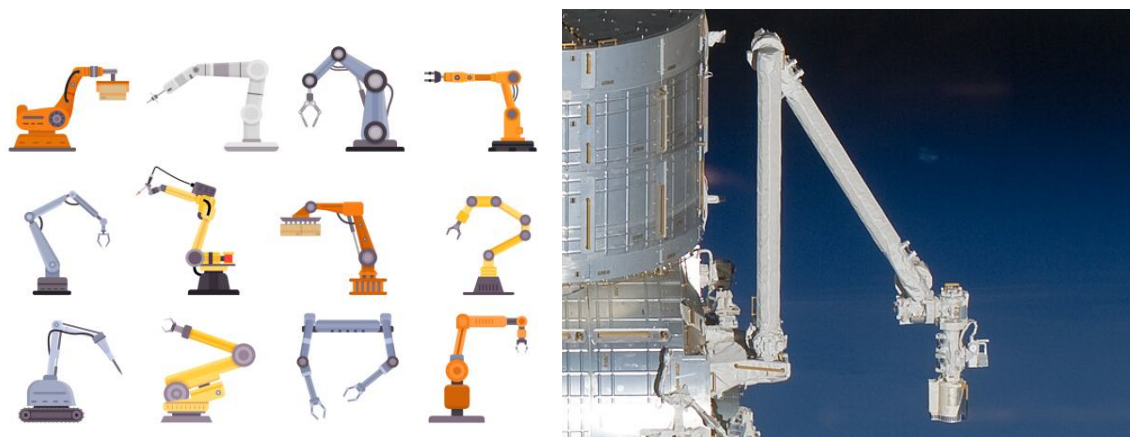
Slika 1.3: Isječci iz Al-Jazarine knjige o automatima: glazbeni automat (gore), sat sa svijećom (dolje)

1.1 Što čini robota

Definirati robota nije trivijalno s obzirom na sličnosti značenja pojmova automat i robot te srodnost robota i automatskih strojeva, ali jedno je sigurno – svaki robot je i automat. Prema Nikoliću (2021.), robota od ostalih strojeva dijele tri karakteristike, a to su senzori pomoću kojih robot dobiva informacije o svijetu oko sebe i na temelju kojih djeluje, računalo odnosno procesor pomoću kojeg dobivene informacije procesira i odlučuje o svojim djelovanjima te izvršni elementi (aktuatori) pomoću kojih robot djeluje na sebe i/ili svijet oko sebe. Nikolić je, potaknut člankom koji je napisala jedna od prethodnih inačica danas već vrlo popularnog jezičnog modela umjetne inteligencije („jezičnog generatora”), napisao članak „Promjene značenja pojma robot u proteklih 100 godina” u kojem je predstavio povijest pojma robot od njegove pojave do danas. Kao prvog robota Nikolić navodi Televox-a, robota koji je djelovao na zvučni podražaj, a korišten je u tvornicama kako bi izvršavao razne zadatke. Međutim, makar je riječ robot već postojala 1925. godine kada je Televox izumljen, nitko ga nije zvao robotom, nego „gospodin Herbert Televox”. Prvi robot s tim nazivom bio je ”Robot Eric” iz 1928. godine koji je na prsima imao napisana slova RUR iz čega možemo zaključiti odakle tvorcima inspiracija za njegovu izradu. Služio je kao „zamjenski čovjek” koji je umjesto vojvode od Yorka otvorio jednu izložbu nakon čega je postao vrlo popularan.

Početne definicije pojma robot uključuju humanoidni oblik kao obavezan faktor u definiranju robota pa tako Enciklopedija Leksikografskog zavoda (1962) kako je navedeno u radu Nikolića (2021) daje definiciju: „robot je automat s ljudskim likom, stroj koji vrši različite, često vrlo komplicirane operacije dirigitiran elektronskim mozgom”. U današnje vrijeme razvoja robotike često se veća pažnja pridaje što boljoj upotrebljivosti robota i jeftinijoj proizvodnji nego skupoj estetici pa tako najkorisniji roboti većinom nisu ljudskog oblika.

Industrijski manipulator, uređaj poznatiji pod nazivom robotska ili mehanička ruka, smatra se pretečom industrijske robotike. Stvoren je četrdesetih godina prošlog stoljeća s namjerom da oponaša rad ljudske ruke u za ljude teškim, opasnim ili čak nemogućim poslovima, prvenstveno za rad u nuklearnim laboratorijima. Danas ima mnoge primjene, na primjer „u industriji i znanstvenim laboratorijima za rad s opasnim kemikalijama, kulturama mikroorganizama i slično, za rad s mikroskopski malenim predmetima u mikrobiologiji, mikroelektronici i preciznoj mehanici, za precizan rad prilikom kirurških zahvata u medicini, za rad na visokim temperaturama, pod morem, u svemiru i drugo.” (Leksikografski zavod M. Krleža, 2013 – 2024). Pojavom manipulatora pojavila se i definicija industrijskog robota koja ne zahtjeva humanoidni oblik. Jedna od najčešćih definicija industrijskog robota je prema Nikoliću (2021) sljedeća: „Robot je reprogramibilni multifunkcijski manipulator, projektiran da prenosi materijale, dijelove, alate i specijalne uređaje kroz različite programirane pokrete u svrhu ispunjavanja različitih zadataka”.



Slika 1.4: Industrijski manipulatori: razni oblici industrijskih manipulatora (lijevo), manipulator na Međunarodnoj svemirskoj postaji (desno)

Ovaj će se rad nadalje koncentrirati na robote u kontekstu obrazovanja, odnosno robote kao alate za učenje programiranja, drugih računalnih vještina i robotike. Takvi roboti, kao i svi drugi, također imaju senzore odnosno ulazne elemente, procesor ili upravljački element i aktuator odnosno izlazne elemente. U nastavku ovog poglavlja ćemo promotriti neke osnovne komponente koje takvi roboti mogu i/ili moraju imati.

Upravljački elementi

Osim samog procesora odnosno kontrolera, robot treba imati izvor energije, a također, treba mu i neki način povezivanja jer je malo vjerojatno da robot može izvršavati operacije potpuno samostalno. Te tri vrste komponenti (kontroler, izvor energije i način povezivanja) spadaju u osnovni hardver robota te ih uglavnom svaki robot posjeduje.

Za samu kontrolu robota je zaslužna neka vrsta procesora, najčešće mikrokontroler koji služi za prikupljanje informacija iz senzora, obradu tih informacija i kontrolu aktuatora. Ti procesori su uglavnom slabi te nisu u mogućnosti izvoditi kompleksne i intenzivne procese kao što je na primjer obrada velikog broja slika već se takvo procesiranje najčešće obavlja na način da robot preko mreže ili nekog drugog komunikacijskog kanala informacije pošalje na drugo računalo koje ima moćniji procesor te se upravo na tom računalu odvija spomenuta obrada podataka. Robot svakako treba imati neki samostalni procesor kako bi prikupljao informacije i upravljao njima.

Robot treba energiju kako bi mogao napraviti išta, a ona može doći iz vanjskih izvora koji su s robotom povezani pomoću kabela ili iz baterijskih izvora koji su danas puno češći s obzirom na razvoj baterijske tehnologije tokom godina. Trenutni standard za napajanje mobilnih uređaja su litij-ionske baterije koje imaju puno veću gustoću energije nego njihovi

prethodnici, a relativno su sigurne za korištenje (ako se koriste na ispravan način). Iako se čini da su baterije koje koriste mobilni telefoni moćne, one ipak u kontekstu robota imaju ograničen učinak. Naime, pokretanje motora koji okreću kotače mobilnih robota zahtijeva puno više energije nego što koriste mobiteli za rad svojih procesora i osvjetljivanje ekrana.

Treći osnovni upravljački element je elektronička komunikacija. Računala uglavnom nisu samostalni uređaji već su tipično spojena na razne periferne uređaje koji znatno pridonose njihovoj funkcionalnosti, a vrlo često su i umrežena s drugim računalima. Kao što su računala povezana s mišem, tipkovnicom, ekranom i drugim perifernim uređajima, tako unutar robota treba postojati povezanost između senzora, aktuatora i mikrokontrolera. Osim unutrašnje povezanosti robota, on uglavnom treba imati mogućnost povezivanja s nekim drugim računalom kako bi ga bilo moguće programirati. Takve veze se također mogu ostvariti i bežično pomoću Wi-Fi-ja ili Bluetootha.

Ulazni elementi

Da bi prikupili informacije o svojoj okolini, roboti trebaju senzore. Ulazni elementi ili senzori su ključni dijelovi jer omogućuju robotima da percipiraju i reagiraju na svoju okolinu. Postoje različiti tipovi senzora koji se koriste u robotici, a neki od najčešćih su:

- **Senzori udaljenosti** mjere udaljenost između robota i objekata u okolini. Mogu biti ultrazvučni, infracrveni ili laserski, ovisno o načinu mjerenja udaljenosti.
- **Senzori dodira** detektiraju fizički kontakt s objektima. Mogu biti u obliku tastera, prekidača ili osjetljivih površina koje reagiraju na pritisak ili dodir.
- **Senzori boje** mogu detektirati boje objekata u svojoj okolini pomoću svjetlosnih senzora ili kamera. Omogućuju robotima da prepoznaju i razlikuju različite boje.
- **Senzori svjetla** mjere količinu svjetlosti u okolini. Koriste se za prepoznavanje različitih svjetlosnih uvjeta i prilagodbu ponašanja robota u skladu s njima.
- **Senzori zvuka** detektiraju zvučne signale u okolini. Omogućuju robotima da reagiraju na zvukove poput glasovnih naredbi, buke ili glazbe.
- **Senzori temperature** mjere temperaturu okoline. Koriste se u robotima za praćenje i regulaciju temperature ili za detekciju promjena temperature u okolini.

Osim ovih senzora koriste se još kompas, žiroskop, akcelerometar, potenciometar, gumbi i mnogi drugi uređaji. Kombiniranjem različitih senzora, roboti mogu percipirati različite aspekte okoline i prilagođavati svoje ponašanje u skladu s tim informacijama.

Izlazni elementi

Da bi djelovali u svojoj okolini, roboti trebaju aktuatora. Izlazni elementi ili aktuatori su ključni dijelovi u robotskim sustavima koji omogućuju robotima da izvrše fizičke radnje ili manipulacije u okolini. Postoje različiti tipovi aktuatora koji se koriste u robotici, a neki od najčešćih su:

- **Motori** su najosnovniji tip aktuatora i koriste se za pokretanje ili rotaciju dijelova robota poput kotača, zglobova ili alata. Postoje različite vrste motora, uključujući istosmjernje motore, koračne motore i servo motore, koji se koriste ovisno o potrebama kretanja i preciznosti.
- **Ekran** je izlazni element koji omogućuje prikazivanje vizualnih informacija korisniku ili okolini. U robotici, ekran se može koristiti za prikazivanje informacija o statusu robota, prikazivanje rezultata senzora ili prikazivanje interaktivnih sučelja za korisnika. Primjeri ekrana u robotici uključuju LCD ekrane, LED matrice ili OLED ekrane.
- **LED svjetla:** LED svjetla su elektronički aktuatori koji emitiraju svjetlost kada su uključeni. Koriste se u robotici za indicaciju statusa, signalizaciju ili osvjetljavanje okoline.
- **Zvučnici** su izlazni elementi koji omogućuju reprodukciju zvuka ili zvučnih signala. U robotici, zvučnici se koriste za emitiranje zvučnih obavijesti, upozorenja ili informacija korisniku ili okolini, a mogu biti u obliku ugrađenih zvučnika, zvučnih modula ili zvučnih sustava s više zvučnika za poboljšanje kvalitete zvuka.
- **Elektromagneti** su aktuatori koji koriste magnetsko polje za privlačenje ili odbijanje objekata. Koriste se u robotici za podizanje, spuštanje ili manipulaciju objektima s feromagnetnim površinama.

Kompleksniji roboti mogu koristiti pneumatske ili hidrauličke cilindre za mehaničke radnje poput podizanja i pomicanja objekata. Pneumatski se cilindri koriste za brze i snažne radnje koje zahtijevaju snagu i brzinu dok se hidraulički koriste za radnje koje zahtijevaju veliku snagu i preciznost. Ovo su samo neki od osnovnih izlaznih elemenata ili aktuatora koje mogu imati roboti. Kombiniranjem različitih aktuatora, roboti mogu izvoditi različite radnje i manipulacije u svojoj okolini, omogućujući im da obavljaju različite zadatke i funkcije.

Softver

Da bi obrađivali signale koje šalju senzori i upravljali aktuatorima, roboti trebaju softver. Softveri za obradu u robotima obuhvaćaju različite vrste programa i alata koji se koriste

za obradu podataka, upravljanje robotima i izvršavanje različitih zadataka. Slijede neki ključni aspekti sustava za obradu u robotima:

- **Operacijski sustav** pruža osnovne funkcije za upravljanje hardverom, izvršavanje aplikacija i komunikaciju s vanjskim sustavima. Roboti mogu koristiti različite operacijske sustave ovisno o njihovim potrebama i mogućnostima. Neke od popularnih opcija uključuju Linux, Windows Embedded, Android i ROS (Robot Operating System).
- **Programski jezici** su ključni za razvoj softvera za robotiku. Postoje različiti tekstualni i blokovski programski jezici koji se koriste ovisno o potrebama projekta, uključujući C/C++, Python, Java, MATLAB, Scratch, Blockly i mnoge druge. Svaki jezik ima svoje prednosti i nedostatke te se koristi za različite svrhe, od niskorazinskih operacija do visokorazinskih algoritama i aplikacija.
- **Biblioteke i okviri** pružaju programerima gotove module, algoritme i funkcije koje olakšavaju razvoj softvera za robotiku. Neki popularni primjeri uključuju ROS (Robot Operating System), OpenCV (Open Source Computer Vision Library), TensorFlow za strojno učenje, Arduino IDE za razvoj Arduino projekata i mnoge druge.
- **Razvojno okruženje (IDE)** je softverski alat koji olakšava razvoj, testiranje i debugiranje softvera za robotiku. Obično uključuje integrirane uređivače koda, alate za upravljanje projektima, debuggere i druge korisne funkcije. Primjeri IDE-a za robotiku uključuju Visual Studio Code, Eclipse, Arduino IDE i mnoge druge.
- **Simulacijski softver** omogućuje programerima da simuliraju ponašanje robota u virtualnom okruženju prije nego što ga implementiraju u stvarnom svijetu. To može uštedjeti vrijeme i resurse te omogućiti ispitivanje i optimizaciju algoritama i kontrola prije nego što se primijene na stvarnom hardveru.

Softveri za obradu u robotima su neophodni za razvoj i upravljanje robotskim sustavima, omogućujući programerima da implementiraju različite algoritme, alate i funkcije kako bi postigli željene funkcionalnosti i performanse robota.

Poglavlje 2

Roboti u obrazovanju

Roboti u obrazovanju bili bi svi oni roboti koji se mogu koristiti u kontekstu obrazovanja. Gruba klasifikacija takvih robota bila bi podjela na one robote koji se koriste kao alati za pomoć pri učenju programiranja, drugih računalnih vještina i robotike te one koji se koriste kao "društveni asistenti" ili učenikovi suradnici koji mu generalno pomažu u učenju nekog školskog gradiva, na primjer jezika. Fokus ovog rada stavljen je na prvu klasu.

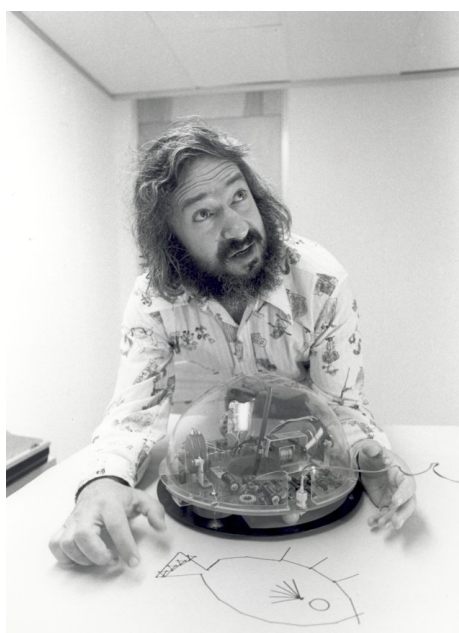
Roboti su u obrazovanju prisutni već mnogo godina. Pošto su se roboti prvo pojavili u tvornicama, obrazovanje se koncentriralo na stvaranje i programiranje industrijskih manipulatora kao dio kurikuluma mehatronike. Kompleksnost tih robota učinila ih je nedostupnima učenicima i studentima koji nisu sa inženjerskih sveučilišnih smjerova.

2.1 Logo i LEGO

Preteča edukacijskih robota kakve znamo danas je programski jezik Logo, prvi programski jezik specifično dizajniran za djecu. Stvorili su ga Seymour Papert i suradnici 1967. s ambicijom da revolucionarizira obrazovanje. Logo se može koristiti u raznim okolinama, a jedna od njih je popularni robot kornjača (*Turtle*). Kornjača je također prisutna i na ekranu računala te se koristi za crtanje raznih oblika i u današnjim učionicama, no ideja robota kornjače nije originalno Papertova već inspirirana radom Williama Grey Waltera. Walter je bio neurofiziolog i robotičar koji je svoje robote kornjače stvorio kako bi bolje razumio rad mozga te da bi demonstrirao svoje teorije o živčanom sustavu u četrdesetim i pedesetim godinama prošlog stoljeća.

Programski jezik Logo konstruiran je tako da ga se uči od poznatog odnosno konkretnog prema nepoznatom ili apstraktnom. Početni koncept koji se uvodi je kretanje u prostoru, nešto što je djeci vrlo blisko jer već imaju razvijen osjećaj za poziciju i smjer kretanja. Apstraktni koncepti poput varijabli uče se tek kasnije, za razliku od klasičnih programskih jezika u kojima je koncept varijable jedna od prvih stvari koja se uvodi. Papert je bio vrlo

angažiran u približavanju matematike, programiranja i računalnih znanosti djeci te je bio vrlo domišljat u načinima kako da to postigne. U svom radu *Mindstorms: Computers, Children, and Powerful Ideas* (1980) opisao je svoju viziju djece koja koriste računala kao alate za učenje. U radu govori kako je temeljna činjenica o učenju: „Sve je lako ako to možete asimilirati u svoju kolekciju modela. Ako ne možete, sve može biti bolno teško.” Bio je vrlo zaintrigiran načinima na koje djeca uče te je osnivač konstrukcionizma, pedagoške teorije nastale pod utjecajem konstruktivizma koju je osnovao Jean Piaget, Papertov svojevremeni suradnik na Sveučilištu u Ženevi. Osim što je bio revolucionar u obrazovanju, Papert je također bio jedan od pionira umjetne inteligencije.



Slika 2.1: Seymour Papert s jednim robotom kornjačom (Molins-Ruano et al., 2017)

Papertovi roboti kornjače bili su nedvojbeno jedni od prvih edukacijskih robota, a to je bio tek početak. Razvijeni su razni roboti kornjače inspirirani Walterovima koji su bili opremljeni motorima za kretanje, mehanizmom za kontroliranje olovke, svjetlima, zvučnikom, sensorima za dodir oko svoje osi i upravljačkom jedinicom. Ovi roboti su se s računalom spajali pomoću žice koja je ograničavala kornjačino kretanje. Programirani su u Logu, a u tu svrhu Papert je razvio i cijelu jednu geometriju koju je nazvao kornjačina geometrija (*Turtle geometry*). Usporedio ju je s Euklidovom i Descartesovom geometrijom rekavši da je Euklidova geometrija logičkog stila, Descartesova algebarskog stila dok je kornjačina geometrija računalnog stila. Također je usporedio samu kornjaču s točkom, jednim od temeljnih koncepata od kojih je sagrađena Euklidova geometrija. Razlika je u

tome da točka ima samo jedno svojstvo – poziciju, dok kornjača ima dva svojstva – poziciju i smjer u kojem gleda. To čini kornjaču malo manje apstraktnom od točke i malo bližom djeci poznatim stvarima. Kornjača gleda u nekom smjeru, isto kao ljudi i životinje. Također, kornjača je dinamična, može se kretati, dok je točka statična. Kornjača na taj način djeci prikriveno približava formalnu matematiku što je jedan od glavnih Papertovih ciljeva. Kao što smo već rekli, kornjača prima naredbe u verziji Loga kojeg Papert u svojoj knjizi naziva kornjačin jezik ili *Turtle talk*. Kornjača može biti u dva oblika, podna kornjača (*floor Turtle*) odnosno robot i laka kornjača (*Light Turtle*) odnosno objekt na ekranu računala. Ova dva oblika će jednako reagirati na Logo naredbe makar su fizički različiti – Papertov mudar način da djeci prikriveno predstavi izomorfizme. Dva osnovna svojstva kornjače, pozicija i smjer gledanja, mogu se promijeniti osnovnim naredbama – FORWARD i BACK (ili kraće FD i BK) mijenjaju kornjačinu poziciju dok ne diraju njen smjer, a LEFT i RIGHT (ili kraće LT i RT) mijenjaju smjer gledanja dok ne diraju poziciju. Uz naredbe treba upisati i broj pa će tako FORWARD 10 pomaknuti kornjaču prema naprijed za 10 kornjačinih koraka dok će RIGHT 30 okrenuti kornjačin smjer za 30 stupnjeva u desno. Jedni od prvih zadataka su crtanje pravilnih geometrijskih oblika kao što su kvadrat, trokut i šesterokut, no prvi izazov postaje nacrtati kružnicu. To se može postići sljedećom naredbom:

```
REPEAT 360 [FORWARD 1 RIGHT 1]
```

Učenike se ohrabruje da pri rješavanju problema pokušaju sebe staviti u kornjačine cipele i otkriju što kornjača treba napraviti da bi riješila zadani problem. Kornjaču se može naučiti da pomoću samo jedne riječi izvrši niz naredbi koje su prethodno definirane što je zapravo metafora za programiranje računala. Prethodnu naredbu se može modificirati u sljedeći program:

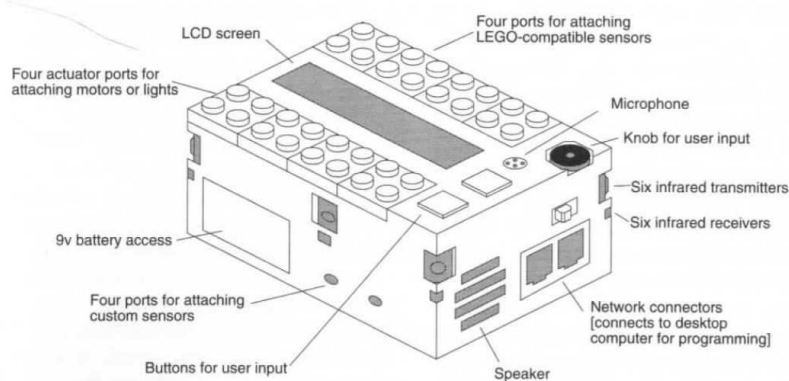
```
TO CIRCLE REPEAT 360 [FORWARD 1 RIGHT 1] END
```

Svaki idući put kad želimo nacrtati kružnicu, možemo to učiniti upisivanjem naredbe CIRCLE koju smo definirali. Ako želimo moći mijenjati dimenziju kružnice, učinit ćemo to idućom modifikacijom:

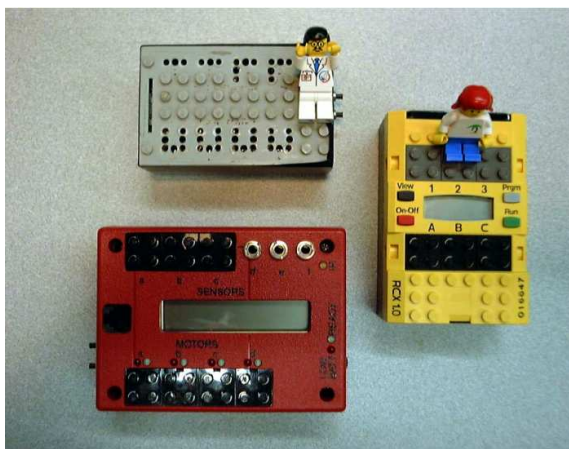
```
TO CIRCLE :D REPEAT 360 [FORWARD :D RIGHT 1] END
```

Na taj način uvodimo varijablu D te ako želimo nacrtati kružnicu kongruentnu prethodnoj, upisat ćemo CIRCLE 1, a ako želimo veću kružnicu možemo upisati CIRCLE 2. Primijetimo da ovim programom zapravo crtamo 360-erokut, ali golom oku on izgleda kao kružnica. Varijabla D predstavlja duljinu stranice tog 360-erokuta, a uz malo matematike može se izračunati koliki je radijus kružnice koju predstavlja taj lik, odnosno opisane kružnice tom 360-erokutu.

Logo je postao vrlo popularan već i prije prvog izdanja Papertove knjige *Mindstorms* 1980., a nakon njezinog izdavanja je pridobio i nastavnike diljem svijeta što je pridonijelo njegovom usponu. Idući veliki korak u razvoju Loga bio je LEGO TC Logo, odnosno sustav koji je povezivao Logo s motorima, svjetlima i senzorima koji su ugrađeni u strojeve izgrađene od LEGO kockica i nekih drugih elemenata. To je omogućavalo djeci da kroz igru uče važne i moćne koncepte programiranja, istovremeno ih povezujući s LEGO kockicama, djeci vrlo bliskim iz predškolskih dana života. Vrlo zanimljiv video na tu temu može se pronaći na YouTube-u pod nazivom: *Seymour Papert Introduces LEGO TC Logo* (poveznica: <https://www.youtube.com/watch?v=BFAX3TDKs00>). Bio je to prvi ikada konstrukcijski komplet za izgradnju robota dostupan široj javnosti. To je bio tek početak LEGO/Logo priče. Idući korak bili su *Technic Control Center* i *DACTA Control Lab*, također LEGO/Logo proizvodi koji su bili programirani pomoću nove verzije Loga zvane *MicroWorlds*. Svi proizvodi do tada morali su biti povezani s računalom pomoću žice da bi primali naredbe, a „*Programmable Brick*” je uveo novu dozu autonomije u svijet edukacijske robotike. Programabilna cigla bila je LEGO/Logo projekt u kojem je izrađeno računalo dovoljno malo da ga LEGO model može nositi, a programirala se na način da je program napisan na računalu preko žice prenesen na ciglu koja se potom mogla odvojiti od računala i samostalno izvoditi operacije. Prva generacija programabilne cigle proizvedena je 1987. godine te je korištena za pokuse provedene u svrhu istraživanja odnosa djece i tehnologije. Bila je uspješan projekt u svrhu istraživanja, ali tek kasnije generacije su postale dovoljno pouzdane da bi bile dio učionice. Druga generacija, zvana „*Red Brick*” proizvedena je između 1994. i 1996. godine u preko sto primjeraka. Temeljno se nije mnogo razlikovala od prve generacije, ali je bila dovoljno stabilna i robusna za korištenje u učionici. I dalje se koristila u eksperimentalne svrhe u tek nekoliko učionica. U tim eksperimentima je razvijen niz aktivnosti u trajanju od jednog semestra te su neki učenici razvijali robote za međunarodno natjecanje.



Slika 2.2: *Red Brick* model (Sargent, R., et al., 1995)



Slika 2.3: Tri generacije programabilnih cigli: prva (gore), druga (dolje), treća (desno) (Martin et al., 2000)

Tokom razvijanja programabilnih cigli stvorena je nova verzija Loga pod nazivom Logo Blocks koji je bio prvi blokovski programski jezik. Logo naredbe su pretvorene u šarene blokove koda koje se mogu pomicati po ekranu i slagati jedne uz druge slično kao puzzle. Treća generacija programabilne cigle je verzija koja je postala komercijalno dostupna, a razvijena je 1998. godine. U suštini se nije mnogo razlikovala od druge generacije, ali je bila manja i imala je tri priključka za senzore te tri za aktuatora (motore). LEGO je prihvatio treću generaciju programabilne cigle, nazvao ju RCX (*Robotic Command eXplorer*) te lansirao brend Mindstorms nazvan po uzoru na Papertovu knjigu. Prvi LEGO Mindstorms set koji je pušten u prodaju 1998. zvao se Robotics Invention System, a sadržavao je RCX programabilnu ciglu, razne LEGO dijelove, dva motora, dva senzora na dodir i jedan senzor svjetla. LEGO je također prihvatio i Logo Blocks okolinu te ju je koristio pod nazivom RCX Code kao službeni programski jezik Mindstorms seta. Osim u RCX kodu, RCX se mogao programirati i u ROBOLAB-u, koji je bio baziran na LabVIEW-u. Bio je to revolucionaran proizvod u području obrazovanja i zabave, istaknut kao alat koji potiče kreativnost i razvoj kritičkog razmišljanja kod korisnika svih dobnih skupina. U narednim godinama, LEGO Mindstorms nastavio je evoluirati, a nova izdanja poput LEGO Mindstorms NXT (lansiran 2006.) i LEGO Mindstorms EV3 (lansiran 2013.) donijela su naprednije komponente i mogućnosti. Set NXT je sadržavao tri servo motora, jedan senzor svjetla, zvuka i udaljenosti, te jedan senzor dodira. NXT 2.0 (lansiran 2009.) je dodao još jedan senzor dodira. Programabilna cigla bila je višestruko moćnija od prve, imala je 3 priključka za motore i četiri za senzore, a povezivanje je bilo moguće i preko Bluetooth-a. Važno je napomenuti da su sve generacije Mindstorms setova uključivale odvojene pakete za maloprodaju i škole koji su ponekad bili poprilično različiti. Na primjer, NXT set za maloprodaju je koristio 6 AA baterija za napajanje, dok je set za škole imao punjivu

litij-ionsku bateriju. Softver koji je došao sa setom zvao se NXT-G (grafičko programsko sučelje), a NXT se mogao programirati i u mnogim drugim programskim jezicima. Iduća generacija LEGO Mindstorms seta je EV3, a njena programabilna cigla radi pod operacijskim sustavom Linux. Povezivanje je, osim Bluetooth-om i USB-om kao na NXT cigli, bilo moguće i preko WiFi-ja te je EV3 cigla imala utor za Micro SD kartice. U EV3 setu se, osim cigle, nalaze dva velika motora, jedan srednji motor, dva senzora dodira, jedan senzor boje, jedan žiroskopski senzor i jedan ultrazvučni senzor. Set za škole je sadržavao manje LEGO kockica nego set za maloprodaju te se dizajn iz maloprodajnog seta nije mogao izgraditi pomoću kockica dobivenih u setu za škole. U vrijeme kada je lansirana prva verzija Mindstormsa, osnovano je natjecanje FIRST LEGO League, rezultat partnerstva LEGO-a i FIRST-a (*For Inspiration and Recognition of Science and Technology*), organizacije pokrenute s ciljem poticanja interesa učenika za inženjering i tehnologiju. LEGO je kasnije razvio konstrukcijske setove LEGO WeDo i LEGO BOOST, namijenjene mladim učenicima. TETRIX je linija konstrukcijskih kompleta za izgradnju robota proizvođača Pitsco Education namijenjena srednjoškolcima. Ima metalne dijelove i robusniji motor za izgradnju moćnijih i kompleksnijih robota, a TETRIX PRIME imao je modul za spajanje na LEGO EV3 ciglu.

Nakon dugogodišnje slave i uspjeha, LEGO je 2022. odlučio umiroviti svoj Mindstorms brend i zamijeniti ga novim. Trenutni proizvod u središtu natjecanja FIRST LEGO League je LEGO Spike, a prethodne komponente Mindstormsa je i dalje moguće koristiti u natjecanju o kojem ćemo govoriti malo kasnije. Postoje dvije verzije LEGO Spike-a trenutno dostupne na tržištu – LEGO Spike Essential za dob od 6 do 9 godina i LEGO Spike Prime za dob od 10 godina na dalje.



Slika 2.4: Sva tri LEGO Mindstorms izdanja (izvor: <https://www.therobotreport.com/lego-discontinues-mindstorms-product-line/>)

2.2 Scratch

Logo Blocks bio je početak nečeg velikog, trenutno prisutnog u osnovnoškolskom obrazovanju diljem svijeta. Godine 2004. pojavilo se novo okruženje za programiranje nazvano Scratch, a od 2007. je dostupno kao aplikacija za rad na računalu. To je blokovski programski jezik nastao pod utjecajem Logo Blocks-a u MIT Media Labu, otkuda je potekao i sam Logo te svi kasniji prethodno spominjani Logo projekti. Scratch je namijenjen djeci koja pomoću njega mogu izrađivati interaktivne priče, igre i animacije na svom računalu te ih dijeliti sa zajednicom na internetu. Od 2017. Scratch je dostupan kao web aplikacija što je uzrokovalo eksponencijalni rast njegove zajednice koja je 2022. godine dosegla preko sto milijuna registriranih korisnika. Originalni osnivači su Mitchel Resnick i David Siegel koji žele osigurati da Scratch ostane besplatan i dostupan svima kako bi djeca diljem svijeta mogla izraziti svoju kreativnost kroz kodiranje. Ciljani dobni raspon korisnika su djeca od osam do šesnaest godina.

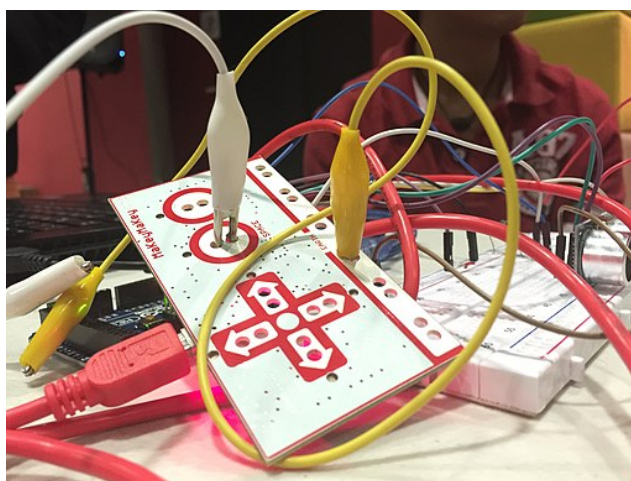
Danas postoji i verzija Scratch-a za mlađu djecu od pet do sedam godina pod nazivom ScratchJr dostupna kao mobilna aplikacija. Na internetskoj stranici www.scratchjr.org dostupni su razni materijali za učenje i podučavanje. U prilogu ovog rada (Prilog 2) nalazi se PDF verzija vodiča za sučelje ScratchJr-a preuzeta sa stranice www.scratchjr.org/learn/interface. Postoje još i vodič za uređivanje boje i vodič sa opisima blokova koji se mogu također preuzeti u PDF-u te videozapisi sa savjetima za korištenje. Ovi materijali spadaju u materijale za učenje. Materijali za podučavanje sadrže niz aktivnosti poredanih od lakših prema težima, resurse za podršku nastavnicima koji nude smjernice i dodatne materijale za korištenje u učionici i izvan nje te ScratchJr Connect, bazu podataka za dijeljenje ScratchJr resursa, lekcija i projekata.

Originalna verzija Scratcha dostupna je na stranici <https://scratch.mit.edu/> preko koje se može pristupiti Scratch web sučelju, raznim vodičima za korištenje Scratch-a, idejama i projektima. Dostupno je i mnogo resursa namijenjenih roditeljima, nastavnicima i programerima. Scratch ima i svoje službene karte koje sadrže kreativne aktivnosti za djecu, a sastoje se od 75 različitih karata koje su također dostupne besplatno online u PDF obliku (na poveznici: <https://resources.scratch.mit.edu/www/cards/en/scratch-cards-all.pdf>). Scratch je vrlo uspješno integriran u današnje učionice pa se tako uvelike koristi i u hrvatskim osnovnim školama u nastavi Informatike. Scratch za svoju veliku globalnu zajednicu organizira razna događanja (virtualna i uživo). Scratch konferencija održava se svake godine u ljetnom terminu, a ove godine slavi svoju desetu obljetnicu. Scratch (online) kamp se također održava ljeti, a sastoji se od tri tjedna u kojima svaki tjedan sa sobom donosi novu temu za projekte. Tema prošlogodišnjeg kampa bila je „*Space, Stars, & Beyond*” (Svemir, zvijezde i još dalje), tema prvog tjedna bila je „*Star Gazers*” (Promatrači zvijezda), tema drugog tjedna „*Mission to Mars*” (Misija na Mars), a tema trećeg tjedna „*Intergalactic Innovators*” (Intergalaktički inovatori). Više

o prošlogodišnjem Scratch kampu može se pronaći u PDF brošuri na sljedećoj poveznici: https://resources.scratch.mit.edu/www/events/en/ScratchLearningResource_ScratchCamp2023.pdf. Osim konferencije i kampa, Scratch također ima svoj tjedan, globalnu virtualnu proslavu Scratch-a i ScratchJr-a, tokom kojeg poziva zajednicu da stvara i dijeli projekte na određenu dnevnu temu. Konačno, dolazimo i do Scratch dana, također namijenjenog proslavi Scratch-a i ScratchJr-a. Scratch dan može organizirati bilo tko bilo koji dan u godini, a više informacija može se pronaći na sljedećoj poveznici: <https://www.scratchfoundation.org/scratch-day>.

Osim programiranja virtualnih likova, Scratch je kompatibilan i s nekim robotima. U sučelju Scratch-a ponuđeni su određeni paketi proširenja uključujući glazbu, olovku, video, alat za pretvaranje teksta u govor, alat za prevođenje jezika te proširenja za nekoliko robota. Nakon uvođenja proširenja, u dijelu sučelja u kojem se nalaze blokovi, pojavljuju se novi blokovi koda vezani uz uvedeno proširenje.

Makey Makey je uređaj za stvaranje i eksperimentiranje koji omogućuje korisnicima da pretvore svakodnevne objekte u interaktivne tipke za računalo. Radi na principu provođenja električne struje kroz objekte koji provode električnost, poput voća, tijesta za modeliranje ili vode. Kada se ti objekti dodirnu ili povežu s Makey Makey uređajem, oni se ponašaju kao tipke na tipkovnici ili mišu, omogućujući korisnicima da ih koriste za kontrolu računalnih aplikacija ili igara. Makey Makey je popularan alat za edukaciju i kreativno eksperimentiranje u različitim kontekstima, uključujući obrazovanje, umjetnost i zabavu, a može se programirati u Scratch-u. Omogućuje korisnicima da stvore jedinstvene interaktivne projekte i eksperimentiraju s osnovnim konceptima elektronike i provođenja električne struje. Također je proizveden na MIT-u.



Slika 2.5: Makey Makey (izvor: https://commons.wikimedia.org/wiki/File:Makey_Makey.jpg)

Go Direct Force and Acceleration je senzor koji se koristi u STEM obrazovanju (*Science, Technology, Engineering, Mathematics* ili, na hrvatskom, znanost, tehnologija, inženjstvo i matematika) za mjerenje sile i ubrzanja. Ovaj senzor je dio linije senzora Go Direct koju proizvodi tvrtka Vernier Software & Technology. Senzor se koristi za prikupljanje podataka u stvarnom vremenu o silama koje djeluju na objekte, kao i o ubrzanju tih objekata, a kompatibilan je sa Scratch-om, no samo kao ulazna komponenta pošto nema aktuatora. Primjena ovog senzora je široka, od eksperimenata iz fizike gdje se mjeri sila koja djeluje na objekte, do eksperimenata iz mehanike gdje se prati ubrzanje objekata. Senzor se može koristiti u raznim nastavnim situacijama, uključujući laboratorijske vježbe, istraživačke projekte i demonstracije na satu.

LEGO proizvodi su također kompatibilni sa Scratchom uključujući LEGO Mindstorms EV3, LEGO BOOST i LEGO Education WeDo 2.0. Još jedan robot, odnosno mikrokontroler ima svoje proširenje unutar Scratch-a, a zove se micro:bit kojeg ćemo pobliže upoznati u nastavku.

2.3 micro:bit

Micro:bit je mikrokontroler lansiran 2015. godine od strane BBC-ja i suradnika s ciljem poticanja digitalne pismenosti među djecom, točnije sedmogodišnjacima u prvoj kampanji. Nakon što je lansiran, osnovana je Obrazovna zaklada Micro:bit (*Micro:bit Educational Foundation*) koja je sada globalna neprofitna organizacija koja upravlja naslijeđem BBC micro:bita i nastavlja ih distribuirati djeci, učiteljima i svima koji žele naučiti programiranje. Ovaj kompaktni mikrokontroler (ili džepno računalo) vrlo je malen, širine pet centimetara i visine četiri centimetra, no opremljen je raznim sensorima i aktuatorima te je posebno dizajniran za učenje osnova programiranja, elektronike i robotike. Micro:bit je opremljen nizom komponenti koje omogućuju raznolike interakcije:

- 25 LED svjetala raspoređenih u matricu dimenzije 5×5 na prednjoj strani pločice, a koja služe za prikaz različitih informacija, slika i animacija.
- Sensori za mjerenje akceleracije i kompas koji omogućuju micro:bit-u da osjeti pokrete i orijentaciju uređaja.
- Dva gumba za kontrolu (A i B) koji omogućuju korisniku interaktivno upravljanje programima i projektima.
- 20 GPIO (*General-Purpose Input/Output*) pinova koji omogućuju povezivanje dodatnih komponenti poput senzora, aktuatora, LED svjetala i drugih elektroničkih uređaja.

- Ugrađeni Bluetooth modul koji omogućuje bežičnu komunikaciju s drugim uređajima poput pametnih telefona, tableta i računala.
- Micro USB utor koji omogućuje povezivanje s računalom i služi za preuzimanje programa te za napajanje.
- Gumb za resetiranje koji omogućuje ponovno pokretanje programa.
- Utor za bateriju koja služi kao alternativni izvor napajanja te omogućuje bežično korištenje micro:bit-a.
- Procesor koji omogućuje sam rad micro:bit-a, a sadrži i temperaturni senzor.



Slika 2.6: Micro:bit v1 prednja i stražnja strana (izvor lijevo: [https://commons.wikimedia.org/wiki/File:BBC_micro_bit_\(2614639942\).png](https://commons.wikimedia.org/wiki/File:BBC_micro_bit_(2614639942).png), desno: [https://commons.wikimedia.org/wiki/File:BBC_micro_bit_\(26212930836\).png](https://commons.wikimedia.org/wiki/File:BBC_micro_bit_(26212930836).png))

Od 2015. do danas prodano je preko šest milijuna micro:bit-a, a često se može naći i u hrvatskim osnovnim školama na nastavi informatike. Trenutno je u prodaji druga generacija – micro:bit v2, koji ima dodatne značajke u odnosu na prvu generaciju: mikrofona i svjetlo u obliku mikrofona koje služi kao indikator mikrofona, logo na prednjoj strani koji je postao osjetljiv na dodir, zvučnik i radio antenu. Na službenim stranicama postoje mnogi materijali kako za učenje tako i za poučavanje (poveznica: <https://microbit.org/>). Kao i za Scratch, postoje razni resursi za nastavnike, a lekcije su podijeljene u dvije dobne skupine – niža dob od sedam do 11 godina i viša dob od 11 do 14 godina.

Osim što je kompatibilan sa Scratch-om, micro:bit ima svoj blokovski programski jezik pod nazivom MakeCode proizvođača Microsoft. MakeCode je blokovski programski jezik

vrlo sličan Scratch-u te dolazi u nekoliko verzija: MakeCode za izradu programa za micro:bit, MakeCode za izradu arkadnih videoigara i MakeCode za Minecraft Education (potrebno je imati Minecraft Education za pristup). Također postoje i MakeCode za Adafruit Circuit Playground Express, MakeCode za LEGO Mindstorms EV3, MakeCode za Chibi Chip i MakeCode za Cue. Adafruit Circuit Playground Express i Chibi Chip su kontroleri slični micro:bit-u, a osim u MakeCode-u, mogu se programirati još i u JavaScript-u. Cue je bio robot dizajniran za pomoć djeci u prelasku s blokovskog u tekstualno programiranje, no više se ne proizvodi.

Osim u Scratch-u i MakeCode-u, micro:bit se može programirati i u JavaScriptu te Pythonu. MakeCode okruženje nudi brzo i jednostavno prebacivanje između tri programska jezika pa je tako moguće program stvoren u blokovskom okruženju jednim klikom prebaciti u kod napisan u Python-u ili u JavaScript-u. Također, u ovom okruženju postoji i simulator micro:bit-a tako da nije čak ni potrebno imati fizički uređaj da bi ga se moglo programirati i vidjeti kako izvodi program.

2.4 Arduino

Još jedno popularno okruženje u području obrazovanja i nastave programiranja je Arduino. To je elektronička platforma otvorenog koda koja je temeljena na jednostavnom hardveru i softveru, prilagođenom za upotrebu svima koji žele kreirati interaktivne projekte. Arduino mikrokontroleri primaju ulazne signale od raznih senzora te ih obrađuju i pretvaraju u izlazne signale koji aktiviraju aktuatora. Programiraju se pomoću Arduino softvera odnosno okruženja (Arduino IDE) u programskom jeziku Arduino koji je baziran na jeziku C++. Arduino je nastao je 2005. u institutu dizajna Ivrea Interaction u Italiji kao jednostavan alat za brzu izradu prototipova, namijenjen učenicima i studentima bez predznanja u području elektronike i programiranja. Danas je, osim u školama, vrlo popularan u svijetu hobista robotike te među umjetnicima i programerima.

Tokom godina je Arduino lansirao preko 100 hardverskih proizvoda, a osim samih mikrokontrolera postoje i drugi dodatci za proširenje mogućnosti samog kontrolera. Trenutno su aktivni mnogi proizvodi, a mnogi su i umirovljeni. Arduino također proizvodi setove u kojima se nalaze razne komponente za stvaranje robotskih uređaja, a neki od trenutno dostupnih setova su Arduino Starter Kit, Arduino Oplà IoT Kit i Arduino Sensor Kit. Arduino kontroleri trenutno dostupni na tržištu su podijeljeni u familije (Nano, MKR, Classic, Mega i druge). Arduino Nano su maleni mikrokontroleri, a dostupni su u mnogim verzijama, od jeftinijih i osnovnijih do moćnijih i naprednijih kontrolera s više značajki. MKR kontroleri su kompaktni i imaju integrirane mogućnosti za međusobno bežično povezivanje (radio modul koji omogućuje povezivanje preko Wi-Fi-ja, Bluetootha i drugih protokola za komunikaciju). Familija Arduino Classic obuhvaća klasične Arduino kontrolere poput legendarnog Arduino UNO, te Arduino Leonardo, Micro, Zero i druge. Arduino

Mega su kontroleri sa moćnijim procesorima za kompleksnije projekte koji zahtjevaju više procesorske snage i više GPIO pinova.

Arduino kontroleri imaju širok spektar primjene i mnoge mogućnosti, ali se samostalno ne mogu smatrati robotima jer sami po sebi nemaju izlazne elemente. Međutim, Arduino ipak ima svoj odjel posvećen obrazovanju pod nazivom Arduino Education koji nudi razna rješenja za učenje i poučavanje. Podijeljen je u tri kategorije: K-12 (što je oznaka u američkom obrazovanju za vrtić i 12 razreda škole – K je oznaka za *kindergarden*), *Higher Education* (više obrazovanje) i *Industry Ready* (spremni za industriju).

K-12 je skup proizvoda prilagođenih američkim kurikulumima, a jedan od proizvoda u fokusu je Arduino Alvik koji je najavljen, ali još uvijek nije lansiran u prodaju. To je kompaktni robot u obliku kvadra koji u sebi krije Arduino Nano ESP32. Sadrži razne senzore: senzor za praćenje linije, senzor udaljenosti, senzor za mjerenje akceleracije, žiroskop, senzor dodira i senzor boje. Također, u njega su ugrađena dva servo motora. Napaja se na punjive baterije koje se pune preko USB-C kabela. Napravljen je tako da se lako može personalizirati raznim dodatcima, a također je kompatibilan sa dizajnima koji se mogu isprintati 3D printerom ili izrezati laserom. Moguće ga je programirati u MicroPythonu, a Arduino u najavi Alvik-a najavljuje i uskoro dostupnu mogućnost blokovskog programiranja. Osim Alvika, Arduino Education obuhvaća razne setove za izradu robotskih uređaja:

- **Science Kit R3** je edukativni set namijenjen učenicima i nastavnicima za provođenje praktičnih eksperimenata iz fizike. Sadrži Arduino Nano i Arduino Science Carrier R3 – dodatak koji mikrokontroleru pruža dodatne mogućnosti, te različite senzore i komponente koje su potrebne za izvođenje eksperimenata, poput senzora za kvalitetu zraka, temperature, vlage, tlaka, svjetla, pokreta, udaljenosti i drugih. Dolazi s priručnikom s detaljnim uputama i lekcijama koje pokrivaju različite teme i koncepte znanosti.
- **Student Kit** je edukativni set namijenjen učenju elektronike i programiranja prikladan za početnike, a posebno dizajniran za učenje na daljinu i od kuće. Opremljen je s 11 aktivnosti koje uvode koncepte poput struje, voltaže i otpora te služe za razvoj vještina rješavanja problema i kritičkog razmišljanja. Također, sadrži hardverske komponente: Arduino UNO R3, matičnu ploču, multimetar, bateriju, žice te razne senzore i aktuatore.
- **Education Starter Kit** je edukativni set namijenjen za učionice. Sadrži iste komponente kao četiri Student Kit-a te je predviđen za do osam učenika.
- **Greenhouse Kit** je edukativni set koji sadrži komponente za održavanje i kontrolu staklenika, a namijenjen je za korištenje u školama. Uz njega dolazi pristup platformi sa sadržajem za učenje i uputama ta postavljanje staklenika. U setu se nalaze: Arduino MKR WiFi mikrokontroler, Arduino MKR IoT Carrier Rev2 dodatak, Grove

Motor Driver za pokretanje motora, senzor kvalitete zraka i CO₂, senzor vlage, senzor temperature, ventilator, vodena pumpa, grove kabeli, LED traka, utikač za struju i još mnogo toga.

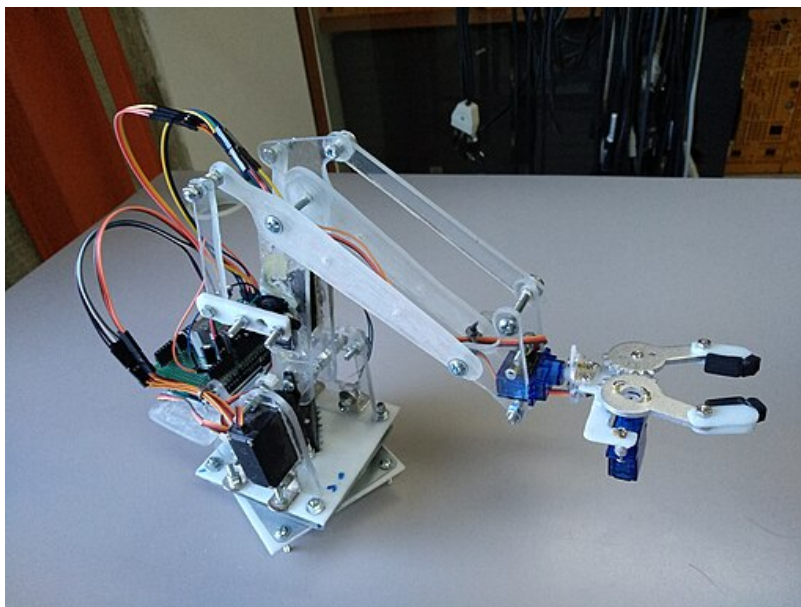
- **Science Kit Physics Lab** je edukativni set razvijen u suradnji s Google-om. Služi za učenje fizike kroz eksperimente u školama. Jedan set pokriva jednog do dva učenika, a sadrži jedan Arduino MKR WiFi kontroler te druge komponente, kabele i senzore.

Osim ovih edukativnih setova, Arduino Education K-12 obuhvaća besplatnu aplikaciju Arduino Science Journal koja je prilagođena i za učionice i za kućno obrazovanje te Arduino Junior Certification mrežni ispit nakon čijeg polaganja Arduino izdaje certifikat. U ispitu se postavlja 25 pitanja u 60 minuta, a tematika pitanja je struja, sintaksa programiranja, elektroničke komponente i Arduino IDE.

Arduino Higher Education i Arduino K-12 se isprepliću u nekim proizvodima kao što su Science Kit R3, Greenhouse Kit, te aplikacija Science Journal, ali uz njih postoje i setovi specifičniji za više obrazovanje:

- **CTC GO!** je edukativni set namijenjen za učenje programiranja i elektronike u srednjoškolskoj učionici. Opremljen je s komponentama dovoljnim za 24 učenika, uključujući osam Arduino UNO kontrolera te druge elemente kao što su baterije, kabeli, senzori, aktuatori i drugo.
- **Explore IoT Kit** je edukativni set namijenjen naprednim učenicima srednje škole i studentima. Služi za učenje o komunikaciji između uređaja, sigurnosti na mreži, prikupljanju, obradi, pohrani te vizualizaciji podataka i sličnim konceptima povezanim uz Internet stvari (*Internet of Things*). Sadrži Arduino MKR WiFi kontroler i Arduino MKR IoT Carrier dodatak razvijen posebno za ovaj set, a ima ugrađene senzore i aktuatora. Sadrži i neke dodatne senzore, žice i mrežni sadržaj za učenje.
- **PLC Starter Kit** je edukativni set za učenje o industrijskoj automatizaciji. PLC ili *Programmable Logic Controller* (programabilni logički kontroler) je tehnologija ključna za industrijsku automatizaciju. Ovaj set je primjeren studentima, a pomaže u spajanju teorije s praksom te priprema studente za svijet rada u industriji i njenom razvoju. Set još nije pušten u prodaju, a imat će svoje vlastito razvojno okruženje Arduino PLC IDE te će ga biti moguće programirati u šest programskih jezika.
- **Braccio ++** je robotska ruka ili industrijski manipulator razvijen za napredne srednjoškolce i studente s ciljem poučavanja o procesima u proizvodnji, dizajnu proizvoda, robotici i automatizaciji. Studenti uče primjene koncepata iz fizike (na primjer pokret, sila, okretni moment, stabilnost, težina tereta i slično) kroz podizanje, pozicioniranje i rotiranje predmeta. Set sadrži sve potrebne komponente za izgraditi robotsku ruku, uključujući Arduino Nano, Arduino Braccio Carrier dodatak, šest

Arduino Smart Servo motora, vijke, matice, opruge, odvijač i još mnogo toga. Napravljen je od recikliranih materijala.



Slika 2.7: Arduino Braccio (izvor lijevo: https://commons.wikimedia.org/wiki/File:Braccio_robotico_con_Arduino_01.jpg)

Osim ovih setova, Arduino Higher Education obuhvaća mrežni ispit Arduino Certification koji je napredniji od prethodno spomenutog mrežnog ispita za Arduino K-12. Također se izdaje certifikat nakon njegovog polaganja, a polaže se na mreži te se sastoji od 36 pitanja u 75 minuta. Tematika pitanja, osim struje, sintakse programiranja, elektroničkih komponenti i Arduino IDE, proširuje se na frekvenciju i radni ciklus, strujne krugove i sheme, programsku logiku i Arduino mikrokontrolere.

Posljednja kategorija Arduino edukacijske linije proizvoda je Industry Ready, a osim nekih već spomenutih proizvoda iz Arduino Higher Education-a, obuhvaća i Engineering Kit temeljen na projektnom učenju pomažući učenicima da razviju inženjerske vještine i temeljne koncepte mehatronike te MATLAB-a i Simulink programiranja.

2.5 Makeblock

Danas postoje mnogi proizvođači koji proizvode razne robote i mikrokontrolere srodne prethodno predstavljenima, a izdvojila bih još privatnu kinesku tehnološku tvrtku Makeblock (osnovanu 2012. godine) koja proizvodi edukacijske setove i komponente za ro-

botiku, elektroniku i programiranje te kompletne edukacijske robote koje nije potrebno sastavljati. Njihovi su proizvodi posebno dizajnirani za edukacijske svrhe, a kompatibilni su za integraciju s drugim platformama i tehnologijama te međusobno povezivanje. Makeblock ima svoje blokovsko programsko okruženje mBlock koje je također vrlo slično Scratch-u, odnosno napravljeno je proširenjem Scratch-a s obzirom da je Scratch program otvorenog koda i kao takav dopušta izradu proširenja.. Proizvodi su podijeljeni u nekoliko skupina ovisno o dobi za koju su primjereni: proizvodi za dob od četiri godine na dalje, za dob od šest godina na dalje, za dob od osam godina na dalje, za dob od 12 godina na dalje te proizvodi za sve dobne skupine. Makeblock proizvodi za dobnu skupinu od četiri godine na dalje uključuju sljedeće robote:

- **Makeblock mTiny** je pametni robot u obliku pande primjeren za djecu predškolske dobi (od četiri do šest godina), a namijenjen za učenje programiranja, matematike, glazbe i jezika. Programira se pomoću kartica koje predstavljaju blokove koda, a funkcioniraju vrlo slično kao ScratchJr. Uz robot dolazi i kontrolna olovka koja služi za čitanje koda, USB kabel, maske za promjenu izgleda robota, mape po kojima se robot može kretati, knjižica s pričama, karte za igru, kartice za kodiranje te zastavice. Robot mTiny se može smatrati modernom verzijom kornjače jer ima mogućnost kretanja te ostavljanja pisanog traga. Osim toga može praviti razne izraze lica, izražavati emocije i ispuštati zvukove te je kompatibilan s LEGO kockicama.
- **Makeblock xLight** je interaktivno svjetlo koje sadrži mikrokontroler, senzor za kontrolu i programabilnu LED traku. Dizajniran je za personaliziranje svjetlosnih efekata noćnog svjetla i razvijanje kreativnosti te vještina programiranja. Podržava razne značajke, kao što su upravljanje gestama, svjetlosni efekti, sinkronizacija s glazbom i druge. Svjetlosna traka se može kontrolirati pomoću mobilne aplikacije, daljinskog upravljača ili senzora za kontrolu, a programira se u mBlock-u.

Druga skupina Makeblock proizvoda, ona za dobnu skupinu od šest godina na dalje, sadrži sljedeće proizvode:

- **Makeblock mBot** jedan je od prvih Makeblock proizvoda, lansiran 2015. godine. Opremljen je raznim elektroničkim, konstrukcijskim i drugim komponentama. Elektroničke komponente uključuju matičnu ploču mCore (koja koristi Arduino Uno kao procesor), senzor za praćenje linije i dva motora. Konstrukcijske komponente uključuju šasiju, dva veća kotača, jedan manji kotač, četiri klina od mesinga, dva RJ25 kabela, osam matica i 25 vijaka. Ostale komponente uključuju odvijač i mini odvijač, USB kabel, držač za AA baterije, daljinski upravljač te dva jastučića za naljepnice na čičak. Dolazi uz razne resurse za učenje i projekte za kodiranje kao što su uvodni projekt *Meet Your mBot* (upoznaj svog mBot-a), *Control mBot like a TV*

set (kontroliraj mBot-a poput televizora) koji služi za upoznavanje s funkcionalnostima daljinskih upravljača, *mBot's Eyes* (mBot-ove oči) koji služi za upoznavanje s ultrazvučnim sensorima te *Make mBot follow a line* (neka mBot slijedi liniju) koji služi za upoznavanje s principima i uporabom senzora za praćenje linije. U paketu dolazi rastavljen tako da ga je prije korištenja prvo potrebno sastaviti što doprinosi upoznavanju djece s elektronikom unutar robota odnosno sensorima, aktuatorima i kontrolerom te raznim drugim komponentama kao što su kotači i drugi konstrukcijski elementi. Kompatibilan je s programskim jezicima mBlock i Arduino te sa LEGO kockicama, a njime se može upravljati i pomoću mobilne aplikacije. Dostupan je edukacijski set i razni paketi dodataka s kojima se mBot može kombinirati:

1. **mBot-S Educational STEM** je edukacijski set dizajniran za škole i grupe. Osim komponentata mBot-a, u paketu dolaze još neke elektroničke komponente (ultrazvučni senzor i matrica LED svjetala), konstrukcijske komponente (razni vijci, matice te držač pod kutem od 120 stupnjeva) i traka za praćenje linije.
 2. **Interactive Light and Sound Robot** je paket dodataka za mBot koji ne sadrži komponente za konstruiranje robota već se sam robot kupuje odvojeno. Sadrži brtvu, ključ za odvijanje matica, dvije grede, ploču pod kutem od 45 stupnjeva, modul sa svjetlosnim sensorom, modul sa RGB LED svjetlima te modul sa zvučnim sensorom.
 3. **Six-legged Robot** je paket dodataka za mBot koji sadrži ključ za odvijanje matica, matice, mnoge vijke i grede te dvije ploče pod kutem od 45 stupnjeva.
 4. **Servo Cat Robot** je paket dodataka za mBot koji sadrži mjedeni klin, osam vijaka, maticu, 6P6C kabel, ključ za odvijanje matica, servo motor, plastični prsten, ploču, držač, spojnicu koja se može rezati, RJ25 adapter te modul sa RGB LED svjetlima.
- **Makeblock mBot Neo** je novija verzija robota mBot koja se ponekad naziva još i mBot2. U usporedbi s prvom verzijom mBot-a, ima nekoliko razlika. Robot mBot2 je kompatibilan s programskim jezicima mBlock i Python, nešto je veći (ali su oba robota iste težine), ima ugrađenu bateriju koja je većeg kapaciteta od baterije na koju radi mBot što znatno produljuje broj radnih sati (od 1-2 radnih sati robota mBot na 5 radnih sati robota mBot Neo), kontroler mCore iz mBota zamijenjen je kontrolerom CyberPi (koji je sam po sebi klasificiran u kasniju dobnu skupinu), veći broj ugrađenih senzora te veći broj senzorskih i aktuatorskih portova. U paketu dolazi jedan CyberPi kontroler, jedan mBot2 štit, jedan ultrazvučni senzor, jedan RGB senzor, jedan motor, dvije glavčine kotača te dvije gume, jedan manji kotač, šasija, dva kabla za motor, dva mBuild kabla od deset centimetara i jedan od 20 centimetara, sveukupno 22 vijka raznih dimenzija, USB kabel, odvijač, traka za praćenje linije te

razni resursi za učenje. Također postoji edukacijski set te razni paketi dodataka za proširenje mogućnosti:

1. **mBot2 Coding Robot** je edukacijski set dizajniran za škole i grupe. Proširen je paketom aktivnosti za izvođenje na nastavi te dodatnim komponentama sličnim kao kod edukacijskog seta mBot.
 2. **Smart World** je paket dodataka za mBot Neo koji ne uključuje komponente potrebne za konstrukciju robota. Sadrži komponente pomoću kojih se mogu stvoriti tri dodatna dizajna: robotska ruka, robotski nosač te robot geodet. U paketu dolaze brojni mehanički dodatci te dva servo motora.
 3. **Smart Camera** je paket dodataka za mBot Neo u kojem dolaze razni konstrukcijski dodatci, pametna kamera mBuild, baterija mBuild, mBuild kabeli, micro USB kabel i set crtičnih kodova. Upravljanje pametnom kamerom odvija se pomoću softvera mBlock 5 i proširenja Smart camera.
 4. Dostupne su i razne dodatne **kutije za kodiranje** u kojima se nalaze primjeri za izradu raznih projekata pomoću robota mBot2.
- **Makeblock Codey Rocky** kompaktan je robot kojeg nije potrebno sastavljati već dolazi u dva dijela: kontroler koji podsjeća na puno moderniju verziju micro:bit-a pod nazivom Codey, te dodatkom za pokretanje robota s kotačima i dodatnim senzorima pod nazivom Rocky. Kompatibilan je s LEGO kockicama te programskim jezicima mBlock i Python. U paketu, osim Codey-a i Rocky-a, dolazi vezica, kartice s bojama, naljepnice i micro USB kabel. Ima mogućnost povezivanja s mobilnim uređajima pomoću Bluetooth-a. Postoji i edukacijski set namijenjen za škole koji u odnosu na osnovni set nema nikakve dodatne komponente.
 - **Bluetooth daljinski Joystick upravljač** kompatibilan sa svim Makeblock proizvodima koji sadrže Bluetooth, a služi za upravljanje njima.

U skupini Makeblock proizvoda za dobnu skupinu od 8 godina na dalje, osim već spomenutog robota mBot Neo, nalaze se sljedeći roboti:

- **Makeblock mBot Mega** jača je verzija robota mBot. Elektroničke komponente od kojih se sastoji su: MegaPi kontroler, modul za Bluetooth komunikaciju, dva upravljača DC motora, četiri DC motora, tri senzora za udaljenost, dva senzora za praćenje linije, senzor koji detektira udarac s prekidačem, dva RGB LED svjetlosna modula te devet prenosnih žica. Od konstrukcijskih komponenti, paket sadrži: štit za tijelo robota, štit za gornji dio robota, držač za šest AA baterija, 68 komada mehaničkih dijelova (odstojnici, manji držači, vijci i matice), četiri veća držača, četiri motorne spojke te četiri kotača. Ostale komponente uključuju: USB kabel, odvijač, nasadni

ključ, traku za praćenje koda te razne resurse za učenje. Programira se u programskim jezicima mBlock, Arduino te Python za korištenje kojeg je potreban još jedan Raspberry Pi.

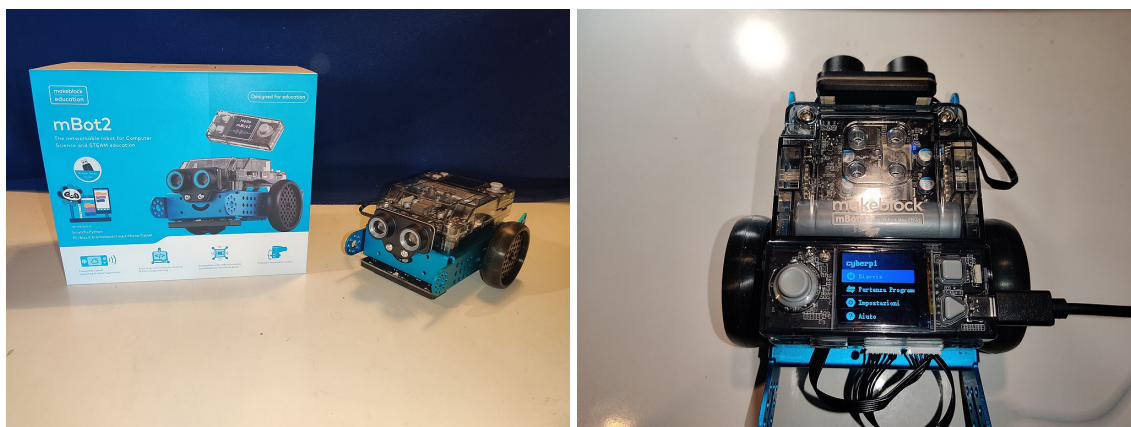
- **Makeblock mBot Ranger** robot je koji se može složiti u tri oblika: robot tenk, trkaći auto i samobalansirajući robot. U paketu se nalaze kontroler Me Auriga, ultrazvučni senzor, senzor za praćenje linije, dva enkoderska motora, tri grede, šest ploča, dva nosača, devet kotača, 44 matice, 72 vijka, tri žice za enkoderski motor, 26 plastičnih odstojnika, dvije gume za kotače, dvije gusjenice za kotače, osam bakrenih čahura, četiri mjedena klina, USB kabel, kabel RJ25, dvostrani ključ, odvijač, mapa za praćenje linije, držač baterija te resursi za učenje. Programira se pomoću programskih jezika mBlock i Arduino. Za mBot Ranger postoji jedan paket dodatka pod nazivom *Laser Sword* (laserski mač), a u paketu se nalaze: tri grede, dvije akrilne ploče za laserski mač, RJ25 adapter, sveukupno 41 vijak, 16 matica, 25 plastičnih odstojnika i jedna LED RGB svjetlosna traka.
- **Makeblock NextMaker** je još jedna verzija robota u čijem paketu dolaze mnogi senzori i aktuatori, a konstrukcijski elementi su mu drvene pločice. U paketu se nalazi: Halocode kontroler, dva upravljača za motore, dva motora, mali „klavir”, zvučnik, dva senzora dometa, upravljač za servo motor, servo motor, dvostruki RGB senzor za boje, Bluetooth dongle, modul napajanja, LED matrični zaslon, klizač, 12 drvenih konstrukcijskih dijelova te 12 setova ostalih komponenti (kotača, vijaka i slično). Kompatibilan je s programskim jezicima mBlock i Python. Makeblock NextMaker ima i svoju mjesečnu pretplatu u kojoj korisnici svaki mjesec dobivaju tri do šest setova materijala za STEM aktivnosti i druge resurse za učenje. Ovi setovi omogućuju izgradnju robota mnogih oblika.

U skupini Makeblock proizvoda za dobnu skupinu od 12 godina nadalje, osim nekih prethodno spomenutih, nalaze se sljedeći proizvodi:

- **CyberPi** je multifunkcionalni kontroler već spomenut kao baza proizvoda mBot Neo. Također se prodaje i kao samostalan proizvod, a u paketu dolaze i džepni štitić za kontroler, USB kabel i resursi za učenje. Osim samog CyberPi-ja, postoje i paketi dodatka za CyberPi: CyberPi Innovation i AI & IoT Creator.
- **mBot Ultimate 10-in-1** je komplet koji sadrži mnoge komponente za izgradnju raznih robotskih projekata, a dostupan je u dvije verzije, za učenike te za škole i grupe. Verzija za učenike sadrži sljedeće elektroničke komponente: kontroler MegaPi, tri DC enkoder motora, četiri MegaPi enkodera koji su ujedno i upravljači za DC enkoderski motor, MegaPi štitić za RJ25, Bluetooth modul, ultrazvučni senzor, senzor za praćenje linije, zatvarač, devet troosnih žiroskopnih modula sa brzinomjerom te

jedan adapter. Konstrukcijski elementi koji se nalaze u paketu su: paket od 55 građivnih dijelova (grede, klizne grede, ploče i slične komponente), paket od 12 spojnica i nosača (spojnice, nosači motora, akrilni nosači, ravni ležajevi), paket od 48 dijelova za kotače, 290 mehaničkih dijelova kao što su vijci i matice te jedna robotska hvataljka. Od ostalih komponenti u paketu se nalaze: držač za šest AA baterija, tri kabla za DC enkoderski motor, USB kabel, dva RJ25 kabla, deset gumica, deset najlonskih veziva, odvijač, dvostrani ključ i šesterokutni ključ. Dijelovi u verziji za škole i grupe se ne razlikuju od onih u verziji za učenike.

U skupini Makeblock proizvoda za sve dobne skupine nalaze se tri proizvoda za škole. To su xTool M1, xTool F1 i xTool P2 – laserski graveri i rezači koji su poprilično skupi. Na svojim internetskim stranicama Makeblock ima dio posvećen edukaciji u kojem se nalaze dodatni resursi za učenje.



Slika 2.8: mBot2 prednja strana s kutijom i gornja strana (izvor lijevo: https://commons.wikimedia.org/wiki/File:Mbot2_with_box.jpg, desno: https://commons.wikimedia.org/wiki/File:CyberPi_Mbot2.jpg)

2.6 Natjecanja iz robotike

Danas postoje još mnogi proizvođači robota sličnih prethodno spominjanima, ali ovdje ćemo se zaustaviti s tom tematikom i prijeći na natjecanja iz robotike koja danas predstavljaju najunčinkovitiji način za popularizaciju robotike. U tim natjecanjima se učenici natječu jedni protiv drugih samostalno ili, puno češće, u timovima, da bi uspješno izvršili niz zadataka. Postoje mnoga lokalna, regionalna i međunarodna natjecanja iz robotike, a u nastavku ćemo promotriti neka najpopularnija.

Hrvatski robotički savez (HROBOS) je krovna neprofitna robotička udruga pravnih osoba u Republici Hrvatskoj koja broji 30 članica. Neki od ciljeva HROBOS-a su razvijanje i promicanje robotike i tehničke kulture u Republici Hrvatskoj te jačanje prepoznatljivosti i podizanje javne svijesti o značaju robotike u društvu. HROBOS je udruga koja organizira natjecanja robotike u Hrvatskoj i sudjeluje u predstavljanju Hrvatske na međunarodnim natjecanjima na kojima hrvatski timovi postižu zavidne rezultate. U Hrvatskoj djeluje i Hrvatsko društvo za robotiku (HDR), osnovano 1994. kao prva i krovna hrvatska nacionalna udruga za robotiku, a danas je dio HROBOS-a. Osim natjecanja iz robotike, HROBOS, HDR i druge udruge i društva u Hrvatskoj organiziraju razne kampove i škole robotike.

Natjecanje FIRST LEGO League smo već spomenuli, a sada ćemo ga pobliže upoznati. FIRST LEGO League (FLL) je međunarodni program natjecanja koji kombinira LEGO kockice i druge LEGO dijelove s inženjerskim izazovima kako bi potaknuo mlade da razvijaju svoje vještine u STEM području. Ovaj je program osnovan s ciljem poticanja interesa mladih za znanost i tehnologiju te promicanja timskog rada, kreativnosti i rješavanja problema. Natjecanje je podijeljeno u tri skupine: FIRST LEGO League Discover (otkrivanje) za dobnu skupinu od četiri do šest godina, FIRST LEGO League Explore (istraživanje) za dobnu skupinu od šest do deset godina te FIRST LEGO League Challenge (izazov) za dobnu skupinu od devet do 16 godina (dob u zadnjoj skupini se može razlikovati u različitim zemljama). U fazi otkrivanja, djeca su uvedena u temeljne koncepte STEM-a rješavajući zabavne izazove i gradeći modele od LEGO kockica. U fazi istraživanja, djeca koriste prethodna znanja iz STEM područja te u timovima izrađuju robote koristeći LEGO SPIKE Essential. U fazi izazova se zapravo krije pravo natjecanje – učenici u timovima izrađuju robote te se pomoću njih natječu u prethodno definiranoj robotskoj igri. FIRST LEGO League svake godine izbacuje novi izazov temeljen na aktualnim znanstvenim temama. Natjecatelji u kolovozu službeno dobiju zadatak i mapu odnosno stazu za natjecanje. Nakon toga slijedi desetak tjedana aktivne pripreme za kvalifikacijsko i završno natjecanje. Pripreme se sastoje od istraživanja, izrade robota i pripremanja materijala za natjecanje. Natjecanje se sastoji od četiri standardne kategorije: Temeljne vrijednosti (*Core Values*) u kojem se ekipe predstavljaju ocjenjivačima, Projekt u kojem ekipe predstavljaju svoj projekt ocjenjivačima u pet minuta, Robotska igra (*Robot Game*) u kojoj ekipe dvije i pol minute igraju igru s ciljem rješavanja što više misija na prethodno određenoj mapi te, posljednja kategorija, Tehnički intervju o dizajnu robota.

Ovo se natjecanje u Hrvatskoj prvi put održalo 2017. godine u organizaciji HROBOS-a na Fakultetu elektrotehnike i računarstva. U Hrvatskoj se od tada natjecanje FIRST LEGO League Challenge održava svake godine. Natjecanje se organizira u suradnji sa slovenskim partnerom *Zavod Super Glavce* koji je i nositelj licence za našu regiju. Mogu se prijaviti učenici od devet do 16 godina i to u ekipama od tri do deset članova koji mogu biti različitih uzrasta. Članovima ekipe se dodjeljuju uloge: kapetan tima, kreativni dizajner,

istraživač, voditelj projekta, komunikator, graditelj, voditelj misije, član zadužen za materijale i programer. Svaka ekipa ima odraslog voditelja koji ne treba imati posebno tehničko predznanje, a ako je potrebna pomoć, uvijek se može zatražiti od mentora stručnjaka koji tada pomažu u pripremi za natjecanje. FIRST LEGO League Explore natjecanje je prvi put održano u Hrvatskoj 2023. godine okupljajući učenike nižih razreda osnovne škole.

RoboCup je međunarodno natjecanje u području robotike osnovano 1997. Održava se jednom godišnje s namjerom promicanja napretka u području robotike i umjetne inteligencije. Glavni cilj RoboCupa je razvijanje autonomnih robota koji će biti sposobni igrati nogomet protiv ljudskih igrača do 2050. godine te istraživanje različitih aspekata robotike, kao što su računalni vid, umjetna inteligencija, mehanika, upravljački sustavi i drugi aspekti. Natjecanje se sastoji od nekoliko liga: RoboCupSoccer (nogomet), RoboCupRescue (spašavanje), RoboCup@Home (kod kuće), RoboCupIndustrial (industrijski) te RoboCupJunior. Svaka od liga dijeli se u nekoliko kategorija pa se natjecanje RoboCup zapravo sastoji od mnogo manjih natjecanja. Ovo natjecanje većinski je namijenjeno starijim učenicima i studentima, osim RoboCupJunior lige koja je namijenjena mlađim učenicima te kojoj je fokus na edukaciji.

RoboCup se u Hrvatskoj prvi put održao 2011. godine u organizaciji HDR-a. Danas su službeni organizatori RoboCup-a u Hrvatskoj HROBOS, HDR, Udruga „Robofreak” i Centar izvrsnosti Varaždinske županije. U Hrvatskoj se održava samo RoboCupJunior liga (RCJ), koja se sastoji od tri kategorije: Soccer (nogomet), OnStage (na pozornici) i Rescue (spašavanje). Kategorija RCJ Soccer sastoji se od natjecanja u kojem mobilni autonomni roboti igraju nogomet „jedan na jedan” ili „dva na dva”. Za dizajniranje, izradu i programiranje robota su zaduženi timovi od po najviše četiri učenika srednjoškolske dobi od 14 do 19 godina starosti. RCJ Soccer dijeli se na nekoliko potkategorija: Soccer Entry, Soccer Lightweight Entry, Soccer Lightweight, Soccer Open te Soccer Small Size League (SSL-Jr). Entry izazovi namjenjeni su početnicima, a roboti igraju „jedan na jedan” za razliku od ostalih u kojima se igra „dva na dva”. Lightweight izazovi imaju ograničenje na težinu robota do 1.1 kilogram. Svi izazovi osim Open igraju se pomoću lopte koja emitira infracrveno svjetlo dok je Open naprednija kategorija u kojoj roboti moraju koristiti sustave računalnog vida kako bi pronašli loptu na terenu, no roboti nemaju ograničenje na težinu što im omogućuje korištenje jačih i težih baterija koje su im potrebne da bi napajale procesor dovoljno jak da može pokretati sustave računalnog vida. SSL-Jr je eksperimentalna liga u kojoj su roboti, teren i računalni sustavi jednaki u svakoj utakmici te su osigurani od strane organizatora. Fokus ove eksperimentalne lige je stavljen na programiranje i strategije utakmica.

Kategorija RCJ OnStage dijeli se na tri potkategorije: OnStage First Steps (prvi koraci – za djecu od pet do devet godina), OnStage Preliminary (početni – za učenike od devet do 19 godina) i OnStage Advanced (napredni – za učenike od 13 do 19 godina). Cilj OnStage natjecanja je kreirati robotsku izvedbu u trajanju od jedne do dvije minute u kojoj robot iz-

vodi neku točku kao na primer ples, kazališnu točku, pripovijedanje ili pokreće umjetničku instalaciju. U kategoriji RCJ Rescue robot ima zadatak spasiti žrtvu, a kategorija se dijeli na sljedeće potkategorije: Line (linija – robot slijedi crnu liniju savladavajući razne izazove i prepreke na putu pokušavajući doći do zone evakuacije i spasiti žrtve), Maze (labirint – robot istražuje labirint, identificira i pomaže spasiti žrtve), CoSpace (dijeljeni prostor – dva virtualna robota su u istom prostoru te pokušavaju skupiti razne predmete i savladati razne izazove) i Rescue Simulation (simulacija spašavanja – robot prolazi kroz virtualni labirint gdje nalazi žrtve i opasnosti te pokušava izraditi mapu labirinta).

Četvrto Europsko RCJ natjecanje održano je u Varaždinu u lipnju 2023. godine na kojem se natjecalo 29 ekipa iz Hrvatske od kojih je sedam osvojilo medalje. U prosincu 2023. godine su hrvatski timovi robotičara ostvarili uspjeh na natjecanju RoboCup Asia Pacific održanom u Južnoj Koreji. Nastupile su tri ekipe; Robofreak, Školska knjiga i ETC Varaždin. Ekipa Robofreak osvojila je drugo mjesto u ukupnom plasmanu u kategoriji Rescue Maze. Ekipa Školska knjiga osvojila je četvrto mjesto u ukupnom plasmanu te posebnu nagradu za najbolju tehničku dokumentaciju u kategoriji Rescue Line. Ekipa ETC Varaždin osvojila je četvrto mjesto u ukupnom plasmanu u kategoriji Soccer LightWeight.

World Robot Olympiad (WRO) je još jedno natjecanje u kojem sudjeluju ekipe iz Hrvatske. Početci WRO natjecanja sežu u 2004. kada je sudjelovalo samo 12 država. Natjecanje se višestruko proširilo pa su 2023. na natjecanju sudjelovale ekipe iz 95 država svijeta i to u ukupno dvadeset tisuća ekipa s preko 55 tisuća natjecatelja. Među njima su bile i hrvatske ekipe od kojih su četiri predstavljale Hrvatsku u finalu: ekipa CHAOS Josipovac 2 osvojila je srebrnu medalju i 14. mjesto u ukupnom poretku u kategoriji ROBOMISSION, ekipa Golden Drop osvojila je srebrnu medalju i deseto mjesto u ukupnom poretku u kategoriji FUTURE INNOVATORS, ekipa Just Here For Cake osvojila je brončanu medalju u kategoriji ROBOMISSION kao i ekipa PrimeRD.

2.7 Tablica

U nastavku ćemo prikazati tablicu koja sadrži robote spomenute u ovom poglavlju te njihove osnovne karakteristike i cijenu. Legenda stupaca u tablici slijedi:

- stupac NAZIV sadrži naziv proizvoda
- stupac DOB sadrži dobne skupine za koje se preporučuje samostalno korištenje
- stupac PROGRAMSKI JEZICI sadrži kompatibilne programske jezike
- stupac CIJENA sadrži cijene preuzete sa službenih stranica proizvoda bez troškova dostave i informacije je li proizvod dostupan u Hrvatskoj

- stupac CIJENA HR sadrži rang cijena po kojima se proizvod može nabaviti u Hrvatskoj s uračunatom dostavom ukoliko se naručuje

Podatci su preuzeti tokom pisanja ovog rada te su podložni promjenama. Tablica je sortirana uzlazno po stupcu DOB.

NAZIV	DOB	PROGRAMSKI JEZICI	CIJENA	CIJENA HR
Makeblock mTiny	4 - 6	Kartice za programiranje	\$133	€186 - 250
Makeblock xLight	4+	mBlock	\$45	-
LEGO SPIKE Essential	6 - 9	Blokovski, Python	\$320	€350 - 400
Makeblock Codey Rocky	6+	mBlock, Python	\$140	€160 - 175
Makeblock mBot Neo	6+	mBlock, Python	\$133	€190 - 200
Makeblock mBot	6+	mBlock, Arduino IDE	\$90	-
micro:bit	7 - 14	Scratch, MakeCode, Python, JavaScript	-	€27 - 50
Makeblock mBot Ranger	8+	mBlock, Arduino IDE	\$150	€240 - 265
Makeblock NextMaker	8+	mBlock, Python	\$150	-
Makey Makey	8+	Scratch	€48	€90 - 150
Makeblock mBot Mega	8+	mBlock, Arduino IDE, Python (uz dodatak)	\$107	-
LEGO SPIKE Prime	10+	Blokovski, Python	\$400	€500 - 550
Arduino Student Kit	11 - 14	Arduino (C++)	€65	€85 - 96
Circuit Playground Express	11+	MakeCode, CircuitPython, Arduino IDE	\$25	€39
Arduino Science Kit R3	11+	Arduino (C++)	€199	€299
Arduino Greenhouse Kit	11+	Arduino (C++)	Na upit	-
Makeblock CyberPi	12+	mBlock, Python	\$70	€95
Chibi Chip	12+	MakeCode, ChibiScript, Arduino IDE	\$35	€163
Arduino CTC GO!	14 - 17	Arduino (C++)	€1500	€1880
Arduino Explore IoT Kit	16+	Arduino (C++)	€125	€162
Arduino Braccio ++	16+	Arduino (C++)	€499	-

Tablica 2.1: Tablica spominjanih robota

Poglavlje 3

Roboti u nastavi

Nastavni predmet Informatika uveden je u hrvatski obrazovni sustav u devedesetim godinama prošlog stoljeća. U redovnom osnovnoškolskom obrazovanju Informatika je trenutno obavezan nastavni predmet u petom i šestom razredu, a u svim ostalim razredima se izvodi kao izborni nastavni predmet te je godišnji broj sati nastave Informatike 70 u svih osam razreda, što odgovara dva nastavna sata tjedno. U srednjoškolskom obrazovanju broj godina učenja obaveznog i izbornog nastavnog predmeta Informatika ovisi o vrsti škole. Prirodoslovno-matematičke gimnazije imaju obaveznu nastavu Informatike sve četiri godine obrazovanja i izvode se u dva programa – A, u kojem se Informatika izvodi dva sata tjedno i B u kojem se izvodi tri sata tjedno. Prirodoslovne gimnazije imaju obveznu Informatiku prva dva razreda, a izbornu druga dva razreda te se nastava održava dva sata tjedno u sva četiri razreda. Jezične, klasične i opće gimnazije imaju obveznu nastavu Informatike samo u jednom razredu i to u drugom razredu jezičnih i klasičnih gimnazija te u prvom razredu općih gimnazija, a nastava se održava dva sata tjedno. U nastavku ćemo promotriti odgojno-obrazovne ishode učenja nastavnog predmeta Informatika koji su povezani s robotima, odnosno ishode u čijim se preporukama spominje riječ robot.

3.1 Kurikulum

Ciljevi nastavnog predmeta Informatika organizirani su u sljedeće četiri domene: Informacije i digitalna tehnologija (domena A), Računalno razmišljanje i programiranje (domena B), Digitalna pismenost i komunikacija (domena C) te e-Društvo (domena D). Roboti se uglavnom većinski vežu za domenu B, Računalno razmišljanje i programiranje te manje za domenu A i C, Informacije i digitalna tehnologija te Digitalna pismenost i komunikacija. Roboti se u kurikulumu spominju već u prvom razredu osnovne škole i to u preporukama za ostvarenje sljedećeg odgojno-obrazovnog ishoda iz domene B:

- **B.1.2**

„Nakon prve godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik prati i prikazuje slijed koraka potrebnih za rješavanje nekoga jednostavnog zadatka.”

Preporuka:

„Prema mogućnostima koristiti se i hardverskim rješenjima za vizualizaciju programiranja (roboti i sl.).”

U drugom razredu osnovne škole roboti se spominju u preporukama za ostvarenje čak dva odgojno-obrazovna ishoda iz domene B:

- **B.2.1**

„...učenik analizira niz uputa koje izvode jednostavan zadatak, ako je potrebno ispravlja pogrešan redosljed.”

Preporuke:

„Na temelju poznatoga zadatka učenik analizira niz uputa predloženih slikom ili riječima, otkriva pogrešan redosljed u uputama i ispravlja ga (niz uputa za zadatke koji su bliski učenicima, igre sakrivanja, davanja uputa za kretanje, vođenje robota labirintom – učenik može glumiti robota).”

„Prema mogućnostima koristiti se i hardverskim rješenjima za vizualizaciju programiranja (roboti i sl.).”

- **B.2.2**

„...učenik stvara niz uputa u kojemu upotrebljava ponavljanje.”

Preporuke:

„Primijeniti pomicanje likova računalom ili bez računala (kornjača, roboti). Koraci rješenja mogu se prikazati i crtežima. Davanje uputa za kretanje, vođenje robota labirintom – učenik može glumiti robota.”

„Prema mogućnostima škole učitelj pokazuje učenicima upravljanje robotom unošenjem različitih naredbi.”

„Prema mogućnostima koristiti se i hardverskim rješenjima za vizualizaciju programiranja (roboti i sl.).”

U trećem razredu osnovne škole roboti se također spominju u preporukama za ostvarenje jednog odgojno-obrazovnog ishoda iz domene B:

- **B.3.1**

„...učenik stvara program korištenjem vizualnoga okruženja u kojemu se koristi slijedom koraka, ponavljanjem i odlukom te uz pomoć učitelja vrednuje svoje rješenje.”

Preporuke:

„Prema mogućnostima škole učitelj pokazuje učenicima upravljanje robotom unošenjem

različitih naredbi.”

„Prema mogućnostima koristiti se i hardverskim rješenjima za vizualizaciju programiranja (roboti i sl.)”

U četvrtom razredu osnovne škole roboti se spominju u razradi i preporukama za ostvarenje jednog odgojno-obrazovnog ishoda iz domene A i u preporukama jednog odgojno-obrazovnog ishoda iz domene B:

- **A.4.2**

„...učenik analizira čimbenike koji razlikuju ljude od strojeva te proučava načine interakcije čovjek – stroj.”

Razrada ishoda:

„Uspoređuje djelovanje ljudi i strojeva (što mogu ljudi, a ne mogu strojevi i obratno). Objasnjava ulogu robota i njihov utjecaj na radna mjesta. Analizira načine interakcije između čovjeka i strojeva.”

Preporuka:

„S pomoću videoprimjera ilustrirati razlike među ljudima i strojevima te primjere upotrebe robota za različite poslove.”

- **B.4.1**

„...učenik stvara program korištenjem vizualnim okruženjem u kojemu se koristi sljedom, ponavljanjem, odlukom i ulaznim vrijednostima.”

Preporuke:

„Prema mogućnostima škole učitelj pokazuje učenicima upravljanje robotom unošenjem različitih naredbi.”

„Prema mogućnostima koristiti se i hardverskim rješenjima za vizualizaciju programiranja (roboti i sl.)”

Vidimo da se kroz prve četiri godine nastave predmeta Informatike znanje gradi na prethodno stečenom, a preporuke za ostvarivanje odgojno-obrazovnih ishoda uvođenjem robota u nastavu su uglavnom poprilično generičke. U petom razredu osnovne škole, robot se spominje u preporukama za ostvarenje dva odgojno-obrazovna ishoda iz domene B, ponovo s generičkim preporukama:

- **B.5.1**

„...učenik se koristi programskim alatom za stvaranje programa u kojemu se koristi ulaznim i izlaznim vrijednostima te ponavljanjem.”

Preporuka:

„Prema mogućnostima koristiti se i hardverskim rješenjima za vizualizaciju programiranja (roboti i sl.)”

- **B.5.2**

„...učenik stvara algoritam za rješavanje jednostavnoga zadatka, provjerava ispravnost algoritma, otkriva i popravljiva greške.”

Preporuka:

„Prema mogućnostima koristiti se i hardverskim rješenjima za vizualizaciju programiranja (roboti i sl.)”

U šestom razredu osnovne škole roboti se spominju u preporukama za ostvarenje dva odgojno-obrazovna ishoda, jednim iz domene B i jednim iz domene C:

- **B.6.1**

„...učenik stvara, prati i preuređuje programe koji sadrže strukture grananja i uvjetnoga ponavljanja te predviđa ponašanje jednostavnih algoritama koji mogu biti prikazani dijagramom, riječima govornoga jezika ili programskim jezikom.”

Preporuka:

„Prema mogućnostima koristiti se i hardverskim rješenjima za vizualizaciju programiranja (roboti i sl.)”

- **C.6.1**

„...učenik izrađuje, objavljuje te predstavlja digitalne sadržaje s pomoću nekoga online i/ili offline programa pri čemu poštuje uvjete korištenja programom te postavke privatnosti.”

Preporuka:

„Koristeći se različitim multimedijским sadržajima (simulacije i videozapisi) i izvorima, pokazati primjere u kojima se računalo koristi modelima inteligentnoga ponašanja kao što su pokretanje robota, govor i prepoznavanje govora, virtualna stvarnost, *Internet of things*.”

U sedmom razredu se roboti ne spominju u odgojno-obrazovnih ishodima, a u osmom razredu se spominju u preporukama za ostvarenje jednog ishoda iz domene C:

- **C.8.3**

„...učenik dizajnira, razvija, objavljuje i predstavlja radove s pomoću sredstava informacijske i komunikacijske tehnologije primjenjujući suradničke aktivnosti.”

Preporuka:

„Odabрати zanimljiva dodatna informatička, njoj slična područja ili područja povezana s informatikom te ih ponuditi kao teme samostalnih/zajedničkih učeničkih radova, npr. pregled mehaničkih računala, razvoj elektroničkih računala, umjetna inteligencija, virtualna stvarnost, učenje igrom, robotika, zanimljive povijesne i recentne priče s temom informatike (Ada Lovelace, Steve Jobs).”

Ovi odgojno obrazovni ishodi sugeriraju uvođenje koncepata programiranja u nastavu vrlo rano i postepeno, po mogućnosti korištenjem robota i sličnih uređaja (na primjer mikrokontrolera).

U Hrvatskoj je trenutno aktivan eksperimentalni program uvođenja novog kurikuluma u osnovne škole pod nazivom „Osnovna škola kao cjelodnevna škola: Uravnotežen, pravedan, učinkovit i održiv sustav odgoja i obrazovanja” (o programu se može više saznati na internetskim stranicama Ministarstva znanosti i obrazovanja Republike Hrvatske). Program donosi mnoge promjene u izvođenje nastave u osnovnoj školi, a pogotovo u nastavnom predmetu Informatike. Mišljenja o programu su vrlo podijeljena. Naime, nastavni predmet pod imenom Informatika prestao bi postojati i zamijenio bi ga predmet naziva Informacijske i digitalne kompetencije. Taj novi nastavni predmet bio bi obavezan u svih osam razreda osnovne škole te bi godišnji broj nastavnih sati bio 35, odnosno jedan sat tjedno. U takvom okruženju bi se stvorio kontinuitet, ali bi se količina nastavnih sati znatno smanjila u odnosu na trenutnu količinu sati obveznog i izbornog nastavnog predmeta Informatika, a isto tako i opseg gradiva koje je potrebno obraditi te broj odgojno-obrazovnih ishoda. Prethodne četiri domene za realiziranje ciljeva nastavnog predmeta Informatika preformulirale bi se u iduće tri domene nastavnog predmeta Informacijske i digitalne kompetencije: Informacijska i medijska pismenost (domena A), Komunikacija, suradnja i sigurnost (domena B) te Računalno razmišljanje, digitalni sadržaji i tehnologije u nastajanju (domena C). Roboti u ovoj podjeli pripadaju samo domeni C. Od trenutnih jedanaest odgojno-obrazovnih ishoda u Informatici u kojima se spominje korištenje robota, u Informacijskim i digitalnim kompetencijama ostalo bi samo tri, a to su:

- **IDK OŠ C.3.1.**

„Učenik analizira niz koraka za rješavanje jednostavnog problema.”

Preporuke:

„otkrivati pogrešan redoslijed u uputama i ispravljati ga (niz uputa za zadatke koji su bliski učenicima, igre sakrivanja, davanje uputa za kretanje, vođenje robota labirintom)”

„koristiti se, prema mogućnostima škole, hardverskim rješenjima za vizualizaciju programiranja robotima ili uređajima sa sensorima”

- **IDK OŠ C.5.1.**

„Učenik stvara program u vizualnome okruženju koristeći se odlukom, ponavljanjem, ulaznim i izlaznim vrijednostima te tipovima podataka.”

Preporuke:

„pokazati, prema mogućnostima škole, upravljanje robotom ili drugim hardverskim rješenjima sa sensorima unošenjem različitih naredbi (*IoT*)”

- **IDK OŠ C.6.1.**

„Učenik stvara programe koji sadržavaju strukture grananja i uvjetnoga ponavlja-

nja.”

Preporuke:

„prema mogućnostima škole koristiti se hardverskim rješenjima za vizualizaciju programiranja korištenjem robota, uređajima sa senzorima (IoT) i sl.”

Vidimo da se roboti spominju samo u odgojno-obrazovnim ishodima za treći, peti i šesti razred, dok su se u Informatici spominjali u ishodima za sve razrede osim sedmog. Odgojno-obrazovni ishod za treći razred nastavnog predmeta Informacijske i digitalne kompetencije kombinacija je prethodno spomenutih ishoda nastave Informatike za prvi i drugi razred. Ishod za peti razred predmeta Informacijske i digitalne kompetencije kombinacija je ishoda za drugi, treći, četvrti i peti razred nastavnog predmeta Informatika, a ishod za šesti razred predmeta Informacijske i digitalne kompetencije je pojednostavljeni ishod za šesti razred nastavnog predmeta Informatika.

Kao što smo već spominjali, izvođenje nastave Informatike u srednjoj školi ovisi o vrsti škole. Posljedično, postoje različiti kurikulumi ovisno o vrstama srednjih škola, točnije, kurikulum se dijeli na tri dijela: Opće, jezične, klasične i prirodoslovne gimnazije koriste jednu verziju kurikuluma, Prirodoslovno-matematičke gimnazije s obaveznom Informatikom sve 4 godine po 70 sati godišnje koriste drugu verziju, a Prirodoslovno-matematičke gimnazije s obaveznom informatikom sve 4 godine po 105 sati godišnje koriste treću verziju. Srednje strukovne škole nisu obuhvaćene ovim kurikulumima. One svoju nastavu također izvode ovisno o vrsti škole, a Informatika je obvezni nastavni predmet uglavnom jednu ili dvije godine. Neke strukovne srednje škole osim Informatike poučavaju i druge nastavne predmete usmjerene ka tehničkom znanju (kao što su na primjer Elektrotehnika ili Računalstvo) pa se takve škole često intenzivnije bave robotima nego gimnazije. U sva tri kurikuluma za gimnazije se robotika spominje samo jednom i to u preporukama sljedećeg odgojno-obrazovnog ishoda koji je jednak za sve vrste gimnazija:

- **C.1.3**

„Nakon prve godine učenja predmeta Informatika u domeni Digitalna pismenost i komunikacija učenik u online okruženju surađuje i radi na projektu”

Preporuka:

„Primjeri tema za istraživanje u sklopu projektnoga zadatka: umjetna inteligencija, robotika, hologrami, prividna stvarnost, mehatronički sustavi, primjene 3D pisača.”

Ovdje možemo vidjeti da ishod zapravo niti ne upućuje na korištenje robota, već na istraživanje robotike. Unatoč tome, kao što smo mogli vidjeti u ranijim poglavljima, roboti bi se itekako mogli koristiti i u srednjim školama pri učenju programiranja, odnosno pri ostvarivanju ishoda iz domene B – Računalno razmišljanje i programiranje. Neki od ishoda iz te domene zahtijevaju da učenici definiraju ulazne i izlazne vrijednosti programa, koriste strukture grananja i ponavljanja, razvijaju i implementiraju algoritme, testiraju programe i slično. Svi ti ishodi mogli bi se povezati s robotima što bi moglo uvelike potaknuti

motivaciju učenika. U većini srednjih škola se na nastavi informatike uči programski jezik Python, a vidjeli smo da ga mnogi kontroleri i roboti podržavaju.

U nastavku ćemo prikazati neke aktivnosti koje se mogu provoditi za ostvarivanje navedenih ishoda.

3.2 Aktivnosti

Krenut ćemo od lakših prema težim aktivnostima te ćemo pratiti ishode. Prisjetimo se ishoda za prva dva razreda osnovne škole:

- **B.1.2** „Učenik prati i prikazuje slijed koraka potrebnih za rješavanje nekoga jednostavnog zadatka.”
- **B.2.1** „Učenik analizira niz uputa koje izvode jednostavan zadatak, ako je potrebno ispravlja pogrešan redoslijed.”
- **B.2.2** „Učenik stvara niz uputa u kojemu upotrebljava ponavljanje.”

Učenici bi za ostvarivanje ovih ishoda mogli koristiti programske jezike ScratchJr i Logo. U prvom bi razredu nastavnik s učenicima mogao obraditi slijed koraka nekog puta, na primjer put kretanja od kuće do škole ili od ulaska u školu do učionice. Na taj način bi nastavnik mogao napraviti uvod u programski jezik Logo. Kroz diskusiju bi nastavnik s učenicima mogao definirati pravila kako će se put zapisivati, a naredbe bi mogle biti sličnog oblika kao što su Logo naredbe. Recimo da bi jedan put mogao izgledati ovako:

RAVNO 10 KORAKA
SKRENI DESNO
RAVNO 25 KORAKA
SKRENI LIJEVO
RAVNO 15 KORAKA

To bi mogao biti put od ulaska u školu do učionice koji je nastavnik s učenicima definirao kroz diskusiju, a potom bi mogao uslijediti zadatak za učenike u kojem oni moraju samostalno definirati put od učionice do ulaska u školu. Za domaću zadaću bi učenici mogli definirati svoj put od kuće do škole koristeći iste naredbe.

Ova aktivnost bi se mogla provesti i korištenjem ScratchJr-a. Naime, na službenim stranicama ScratchJr-a postoje PDF dokumenti za ispis u kojima se nalaze sličice blokova ScratchJr koda (<https://www.scratchjr.org/teach/resources>). Nastavnik bi mogao isprintati velike blokove za korištenje na ploči te male blokove koje bi podijelio učenicima prilikom izvođenja ove aktivnosti.

Ova aktivnost mogla bi biti uvod u programske jezike Logo i Scratch. Naime, prilikom pokazivanja kretanja kornjače po ekranu upisivanjem Logo naredbi, takav način zadavanja naredbi bio bi već poznat učenicima tako da bi ga mogli lakše savladati. Također, ScratchJr blokovi mogu se povezati sa Scratch blokovima u kasnijim aktivnostima.

U trećem razredu bi se mogao uvesti blokovski programski jezik Scratch u sklopu sljedećeg ishoda:

- **B.3.1** „Učenik stvara program korištenjem vizualnoga okruženja u kojemu se koristi slijedom koraka, ponavljanjem i odlukom te uz pomoć učitelja vrednuje svoje rješenje.”

Učenici bi mogli početi koristiti Scratch i programirati likove na ekranu te nakon što savladaju korištenje blokovima, mogli bi prijeći na micro:bit ili Codey Rocky. U početku bi se mogle raditi jednostavne animacije korištenjem petlji. Ovdje bi se mogle iskoristiti Scratch karte spominjane u prethodnom poglavlju. Učenicima ne bi trebalo biti teško savladati zadatke iako su karte na engleskom jeziku jer, uz tekst, također imaju i slikovne upute. Takve aktivnosti bi se mogle povezati i s učenjem stranih jezika ili bi ih nastavnik mogao iskoristiti kao inspiraciju za kreiranje aktivnosti na hrvatskom jeziku.

U četvrtom razredu dolazimo do uvođenja ulaznih vrijednosti u sklopu sljedećeg ishoda:

- **B.4.1** „Učenik stvara program korištenjem vizualnim okruženjem u kojemu se koristi slijedom, ponavljanjem, odlukom i ulaznim vrijednostima.”

Za ostvarivanje ovog ishoda bilo bi vrlo vrijedno iskoristiti robote s obzirom da roboti imaju senzore te preko njih primaju ulazne vrijednosti. Učenici bi mogli isprogramirati svoje kontrolere micro:bit te ih potom koristiti za igru.

Zadatak: bacanje kocke

Učenici su prethodno upoznati s blokovskim programiranjem i kontrolerom micro:bit.

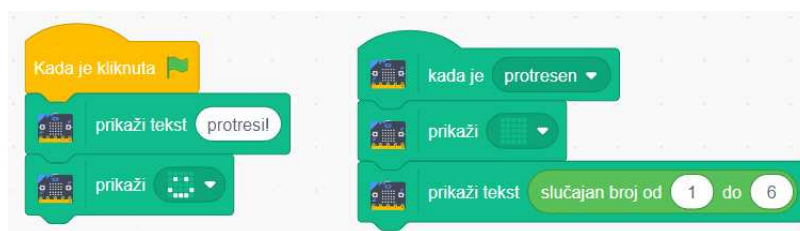
Zadatak: Isprogramirati micro:bit da prvo pokaže uputu „Protresi”, kada ga se protrese, na ekranu prikaže nasumičan broj od jedan do šest.

Rješenje:

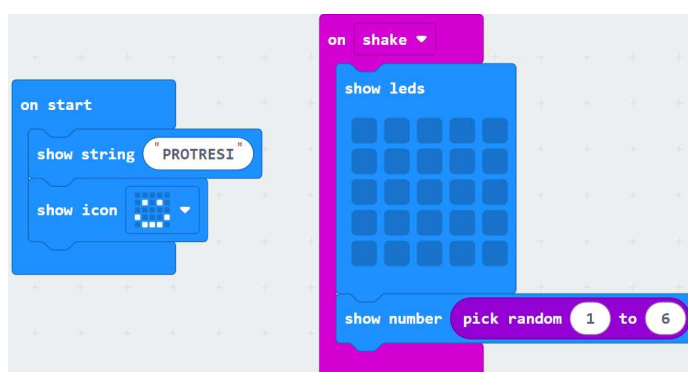
Za rješenje zadatka potrebno je koristiti osnovne blokove za odluke bazirane na ulaznim vrijednostima, prikazivanje na ekranu i odabir slučajnog broja. Neka dio učenika koristi Scratch, a ostali MakeCode okruženje. U nastavku, na slici 3.1, možemo vidjeti rješenje zadatka u Scratch-u, a na slici 3.2 rješenje u MakeCode-u.

Nakon rješavanja komentirati sličnosti i razlike u rješenjima te mogućnosti poboljšanja programa. Jedina logička razlika je da Scratch nema blok koji bi odgovarao „on start” bloku iz MakeCode-a. To znači da ako bi micro:bit bio odvojen od računala, nikada ne

bismo mogli pokazati početnu poruku s uputom. Također, ovo rješenje ima poboljšanje kojeg se učenici možda ne bi dosjetili iz prve, već tek nakon testiranja programa. Naime, ako dvaput za redom dobijemo isti broj to nećemo vidjeti osim ako ne postavimo prikaz praznog ekrana između.



Slika 3.1: Rješenje zadatka bacanje kocke u Scratch-u.



Slika 3.2: Rješenje zadatka bacanje kocke u MakeCode-u.

Nakon rješavanja zadatka, učenici mogu koristiti micro:bit umjesto kocke kako bi igrali neku društvenu igru ili „par – nepar” (tako se učenike može motivirati da što prije riješe zadatak da se mogu igrati).

U petom razredu se prvi put uvode varijable, tako da je to idealan trenutak za obraditi s učenicima zadatak sličan prethodnom. Ovaj put ćemo pospremiti dobiveni slučajni broj u varijablu kako bi na temelju njega odredili izlaznu vrijednost.

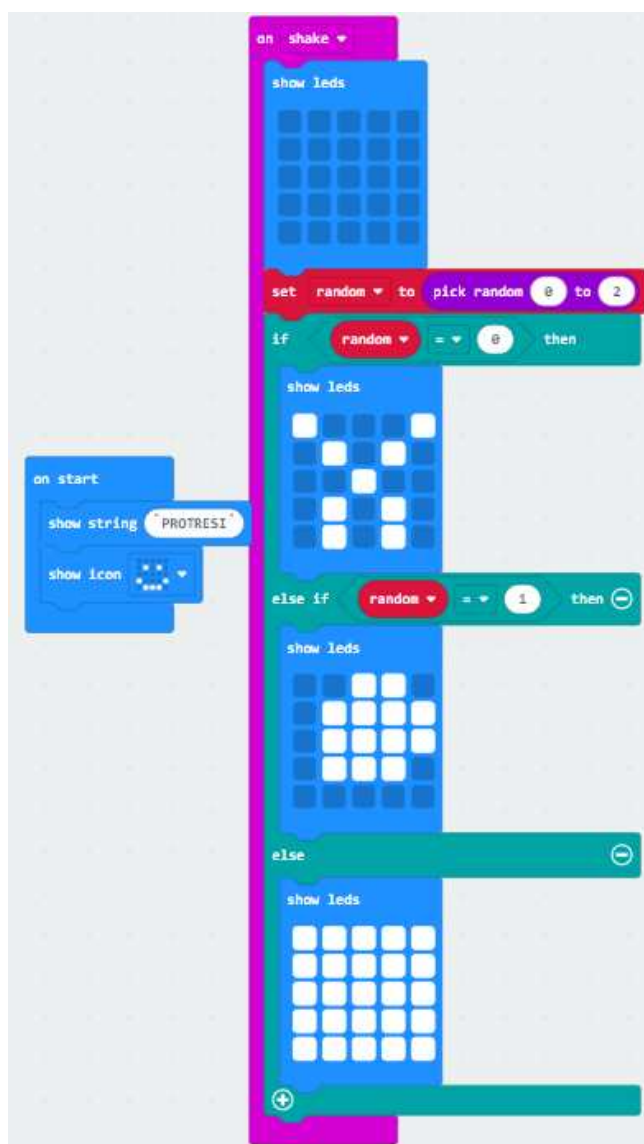
Zadatak: kamen, škare, papir

Učenici su upoznati s konceptom varijable.

Zadatak: Isprogramirati micro:bit da se pomoću njega može igrati igra kamen, škare, papir.

Rješenje:

Za rješenje zadatka potrebno je koristiti osnovne blokove za odluke bazirane na ulaznim vrijednostima, prikazivanje na ekranu te varijable i odabir slučajnog broja. Zadatak se može riješiti pomoću Scratch ili MakeCode okruženja.



Slika 3.3: Rješenje zadatka kamen, škare, papir u MakeCode-u.

Pomoću ovog programa učenici mogu igrati kamen, škare, papir jedni protiv drugih. Učenicima se može pokazati Python kod ovog programa:

```

def on_gesture_shake():
    global random
    basic.show_leds("""
        . . . . .
        . . . . .
        . . . . .
        . . . . .
        . . . . .
        """)
    random = randint(0, 2)
    if random == 0:
        basic.show_leds("""
            # . . . #
            . # . # .
            . . # . .
            . # . # .
            . # . # .
            """)
    elif random == 1:
        basic.show_leds("""
            . . # # .
            . # # # #
            . # # # #
            . # # # .
            . . . . .
            """)
    else:
        basic.show_leds("""
            # # # # #
            # # # # #
            # # # # #
            # # # # #
            # # # # #
            """)
input.on_gesture(Gesture.SHAKE, on_gesture_shake)

random = 0
basic.show_string("PROTRESI")
basic.show_icon(IconNames.HAPPY)

```

Analiziranjem i uspoređivanjem ovog koda s programom kojeg su napisali u blokovima, učenici će povezati Python s kojim se do sad nisu susretali, s nečime što im je poznato od prije. Učenici sada mogu nastaviti programirati micro:bit u Pythonu, a za početak mogu pomoću MakeCode okruženja umetati blokove Python koda u program na sličan način kao što su već naučili ubacivati komade blokovskog Scratch-a. U višim razredima se roboti mogu nastaviti programirati pomoću programskog jezika Python te tako zadržati motivaciju učenika pri programiranju kompleksnijih programa.

Prethodni zadaci su se mogli riješiti i pomoću robota Codey Rocky, a za idući zadatak nam je on potreban. U srednjoj školi bi se učenicima moglo pokazati jedno vrlo zanimljivo proširenje u mBlock okruženju pod nazivom *Teachable Machine* (stroj kojeg se može naučiti). To je službeno proširenje računalnog vida koje je razvio mBlock. Zadatak koji slijedi smo razvile kolegica Karla Juranović i ja u sklopu kolegija Programski jezici i oprema za nastavu programiranja u školama.

Zadatak: kamen, škare, papir i strojno učenje

Za rješavanje zadatka potrebni su Codey Rocky i računalo s kamerom.

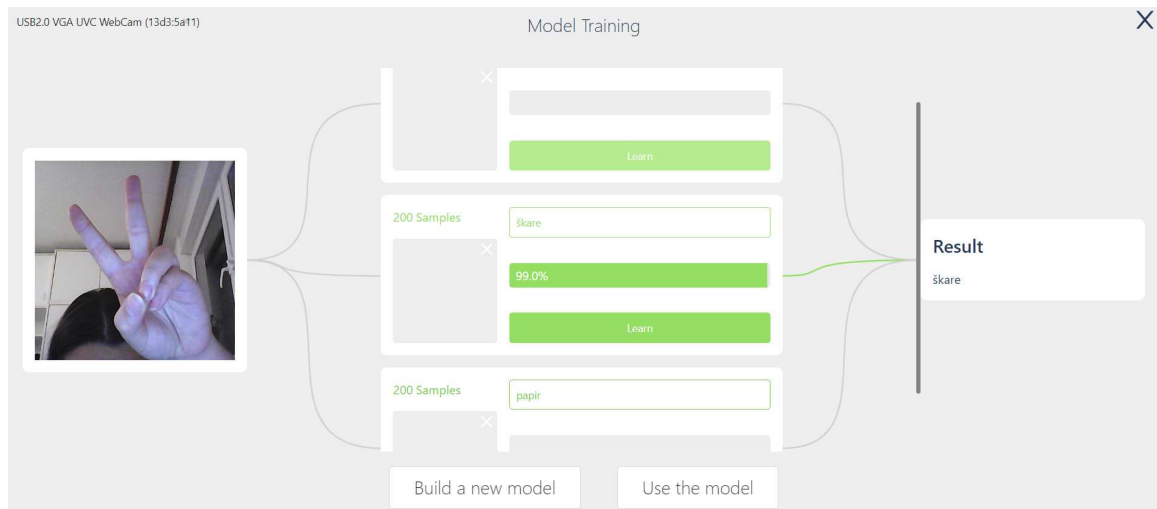
Zadatak: U mBlock okruženju isprogramirajte Codey Rocky robota da igra igru kamen, škare, papir protiv vas. Upute za rješavanje zadatka:

1. Pomoću proširenja *Teachable Machine* istrenirajte model za prepoznavanje na način da prepozna kada pokazujete kamen, kada škare, a kada papir.
2. Isprogramirajte Pandu da na temelju rezultata prepoznavanja emitira poruku.
3. Isprogramirajte robota Codey Rocky da na temelju slučajne varijable odabere kamen, škare ili papir, te u kombinaciji s primljenom porukom prikaže određenu emociju.

Ako je Codey pobijedio, onda je ponosan, ako je izgubio, onda je tužan, a ako je neriješeno, onda je začuđen.

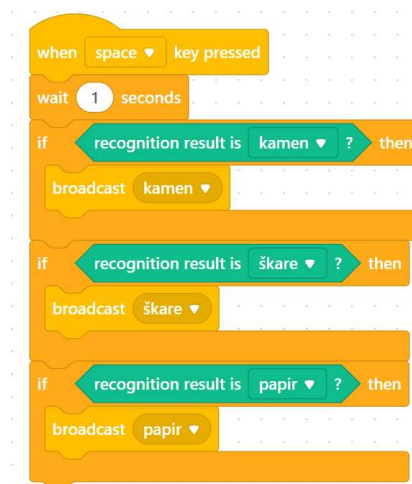
Rješenje:

1. Nakon uvođenja proširenja *Teachable Machine* treba otvoriti sučelje za treniranje modela. U sučelju se s lijeve strane nalazi ulazni element iz kamere (slika), u sredini se nalaze kategorije koje treba definirati, a desno se nalazi rezultat prepoznavanja. U slučaju ovog zadatka će kategorije biti kamen, škare i papir. Pritiskom na gumb *Learn* kod svake kategorije započinjemo proces treniranja. Model se trenira na fotografijama koje dolaze iz kamere računala. Što više fotografija ima, to će model bolje raditi. Na slici 3.4 je prikazano sučelje istreniranog modela koji je prepoznao škare. Nakon što smo istrenirali model, prelazimo na programiranje Pande.



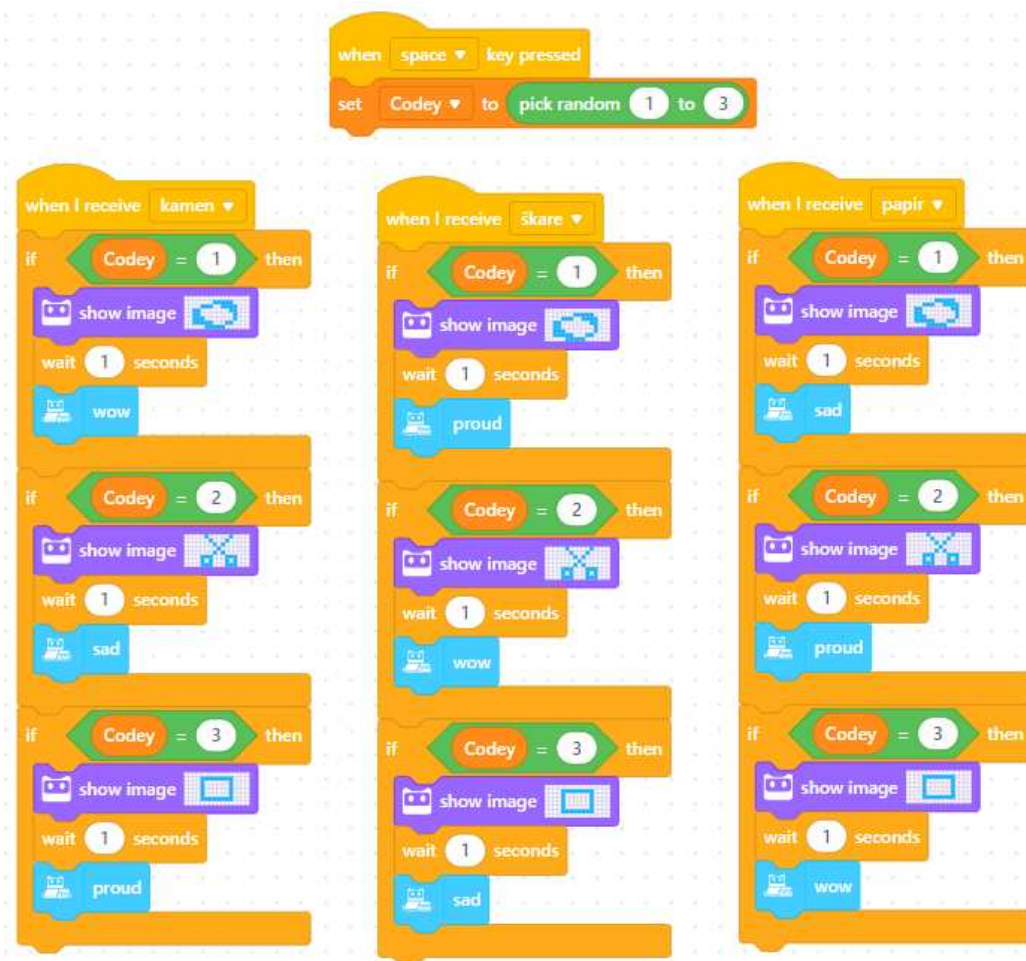
Slika 3.4: Sučelje za treniranje modela (prepoznate su škare)

2. Panda u ovom slučaju ima funkciju posrednika. Koristimo ju zato što sam robot Co-dey Rocky nema mogućnost primiti vrijednost direktno iz modela, već to može samo Panda. Pandu programiramo tako (slika 3.5) da kada iz modela primi informaciju da je prepoznao, na primjer, škare, šalje poruku „škare” pomoću *broadcast* naredbe za emitiranje. U ovom zadatku još moramo definirati i događaj koji kroz sva tri isprogramirana sučelja (model, Panda i robot) istovremeno pokreće program, na primjer, pritisak na neku tipku na tipkovnici računala.



Slika 3.5: Program u sučelju lika Pande

3. Preostaje samo isprogramirati robota. On prilikom pokretanja programa mora generirati slučajnu vrijednost koja predstavlja robotov odabir za kamen, škarice ili papir. Potom, na temelju poruke koju prima od Pande preko naredbe *broadcast* i slučajne vrijednosti koja je na početku programa generirana, robot odlučuje o svojim daljnjim koracima. Tri su slučaja za robotov odabir te su tri moguće poruke što čini devet mogućih scenarija. Na slici 3.6 je prikazan program u sučelju robota Codey Rocky.



Slika 3.6: Program u sučelju robota Codey Rocky

Ova aktivnost zasigurno može zainteresirati i starije učenike te im može prenijeti vrijedno znanje o strojnom učenju i objektnom programiranju, makar su robot i blokovsko progra-

miranje namijenjeni za mlađe učenike. U nastavku ćemo promotriti kako stariji učenici reagiraju na robota Codey Rocky uz niz jednostavnijih zadataka.

3.3 Roboti u MIOC-u

U sklopu kolegija Metodička praksa iz informatike u srednjoj školi, održala sam u zagrebačkoj XV. gimnaziji (MIOC) pod mentorstvom Ernesta Wendlinga i Gorana Igalyja sate na kojima su učenici programirali robote Codey Rocky u blokovskom programskom jeziku mBlock. Cilj održavanja ovih sati bio je prvenstveno promotriti reakcije učenika na pojavu robota u njihovoj učionici i prikupiti opažanja o angažiranosti učenika u takvom nastavnom okruženju. Održan je blok sat u dvije odvojene grupe, prva grupa je bila 3. razred, a druga grupa 2. razred te je u grupama bilo po desetak učenika. Početne, a i završne reakcije bile su vrlo pozitivne iako su ovi roboti namijenjeni mlađim učenicima (osnovna škola). Učenike sam prvo uvela u blokovski programski jezik mBlock, a potom u robota Codey Rockyja nakon čega sam im zadala dva nastavna listića, prvi s tri jednostavna zadatka, a drugi s tri malo kompliciranija zadatka, no svi zadaci su bili vrlo lako izvedivi za njihovu dob i znanje programiranja. Prije nego što sam im podijelila drugi nastavni listić sam im pokazala Neuron – konstrukcijski kit za sastavljanje robota po vlastitim preferencijama koji je kompatibilan s Codeyem te se na taj način mogu proširiti Codeyevim mogućnostima. Cilj je bio vidjeti hoće li učenici imati svoje ideje i uspjeti doći do kreativnih rješenja ili će slijepo pratiti zadatke s listića i smatrati Codeya igračkom. Zanimljivo je da su stariji učenici reagirali bolje od mlađih te se više udubili u rad i shvatili potencijal Codeya dok su ga mlađi više vidjeli kao igračku. Makar su gotovo svi učenici zadatke smatrali prelaganima (12 od 15 odazvanika na anketu), također su gotovo jednoglasno odgovorili da bi htjeli imati više sati s robotima u nastavi. Na pitanje „Biste li voljeli imati više sati u školi u kojima biste programirali robote?” 9 učenika odgovorilo je „Da, svakako”, 5 učenika odgovorilo je „Da, ali samo ponekad”, 1 učenik je odgovorio „Ne znam” te nitko nije negirao zainteresiranost (ponuđeni odgovori bili su još „Ne, radije bih se fokusirao/la na druge predmete” i „Ne, uopće me ne zanima programiranje robota”). Makar je većina učenika odgovorila da im je Codey zanimljiv samo kao igračka (9 od 15, ostalih 6 odgovorilo je „Izuzetno je zanimljiv i koristan”), na pitanje „Biste li voljeli učiti Python uz pomoć robota poput Codey Rockyja?” 12 učenika odgovorilo je „Da, svakako”, a 3 učenika odgovorila su „Možda, ovisi o izvedbi” (ponuđeni odgovori bili su još „Ne znam” i „Ne zanima me to”). Učenike sam u anketi upitala jesu li vidjeli prije robota sličnog ovom, a odgovori su bili vrlo raznoliki. Petoro učenika nije nikad vidjelo sličnog robota, a svi ostali, njih deset, su se već susreli s robotima te su podijelili svoja iskustva. Troje učenika reklo je da je vidjelo robote u osnovnoj školi – jedan je radio s micro:bitom, drugi s arduinom, a treći je rekao da je u osnovnoj školi imao radionicu robotike, ali nije specificirao koje je robote vidio. Dvoje učenika vidjelo je robote u privatnom životu – jedan učenik rekao je

da ima prijatelja koji se bavi robotikom, a drugi je rekao da mu tata radi s robotima. Petoro učenika vidjelo je robote u izvanškolskim aktivnostima – jedan je rekao da je vidio LEGO robote na FER-u, drugi je 2019. godine bio na HSIN (Hrvatski savez informatičara) kampu na Krku, a također ima takvog robota (pretpostavljam Codey Rocky-a) kod kuće kojeg je dobio za rođendan prije pet godina, treći učenik bio je na Erasmus projektu i dječjim radionicama gdje je vidio LEGO Mindstorms i LEGO Spike, četvrti je bio na Scratch kampu u Algebri, a peti je bio na radionicama i na FLL (FIRST LEGO League) natjecanju te ima sličnog robota kod kuće.

U prilogu ovog rada (Prilog 1) se nalazi priprema za izvođenje tog blok sata.

Poglavlje 4

Prilozi

4.1 Prilog 1

Priprema za izvođenje nastave

Predmetni nastavnik: Ernest Wendling

Nastavni predmet: Informatika

Razredni odjel: 3.

Datum: 26. 4. 2023.

Tema: Objektno programiranje robota u blokovskom programskom jeziku

Ishodi učenja teme:

Učenici će programirati robota u blokovskom programskom jeziku.

Međupredmetne teme:

- **pod C. 5. 1.** sudjeluje u projektu ili proizvodnji od ideje do realizacije
- **ikt A.5.3.** preuzima odgovornost za vlastitu sigurnost u digitalnome okružju i izgradnju digitalnoga identiteta
- **uku C.4/5.3.** iskazuje interes za različita područja, preuzima odgovornost za svoje učenje i ustraje u učenju

Tip sata: obrada novog gradiva, individualni rad

Literatura:

- <https://www.makeblock.com/pages/codey-rocky-emo-robot>
- <https://www.mblock.cc/en/>

Tijek nastavnog sata (artikulacija)

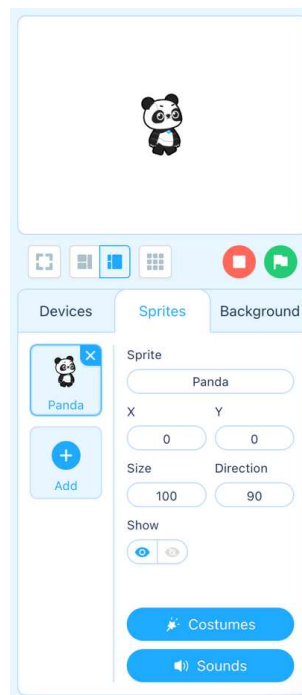
Aktivnost 1: (15-20 minuta) Uvod

Učenicima govorim da ćemo danas programirati robota, odnosno konkretnije Codey Rockyja. Pitam učenike jesu li ikada vidjeli takvog ili sličnog robota i jesu li ikad čuli za mBlock, Scratch, ili MakeCode, odnosno neki drugi blokovski programski jezik.

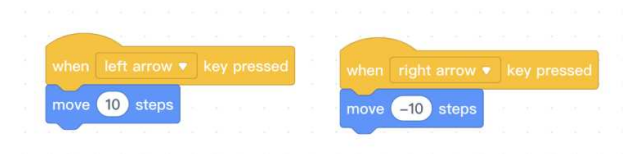
Mi ćemo danas raditi u mBlocku pa im govorim da otvore mBlock na svojim računalima te da ćemo se prvo malo upoznati s mBlockom. Učenici trebaju otvoriti stranicu <https://mblock.makeblock.com/> na kojoj potom moraju kliknuti na gumb „Code with blocks“.

Za početak na ekranu vidimo pandu (slika desno), ona je jedan od *spriteova*, odnosno likova. Ona ima svoj položaj, odnosno koordinate na ekranu, možemo ju pomaknuti mišem, a možemo i upisati koordinate na koje želimo da se pomakne. Osim likova, u izborniku imamo *devices* (uređaji) i *background* (pozadina). U *devices* se nalaze razni roboti, pa i naš Codey, ali o tome malo kasnije. Možemo promijeniti pozadinu koja se nalazi iza pande. Pitam učenike da izaberu koju ćemo pozadinu staviti.

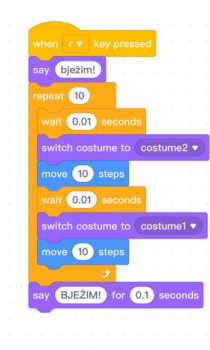
U ovom slučaju su panda i odabrana pozadina objekti, panda je u klasi likovi (*sprites*), a pozadine su druga klasa. Naš robot je u klasi uređaji. Sada ćemo prijeći na programiranje pande.



Primjer 1: Panda ide lijevo desno pomoću tipkovnice (koristimo blokove iz polja EVENTS i MOTION – događaji i pokreti)




Primjer 2: Panda trči. (koristimo blokove iz polja LOOKS i CONTROL – izgled i kostimi, petlje)



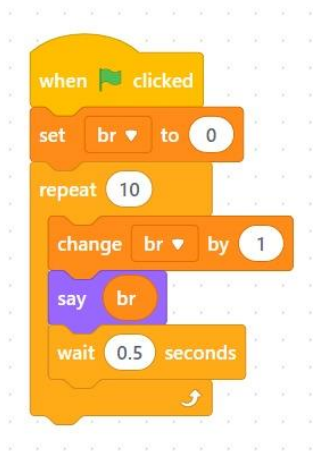
Sada ćemo dodati još jednu značajku našem programu.

Idemo za početak vidjeti kako naša panda može izgledati, otvorimo *costumes* dio. Ovdje vidimo da naš lik ima dva definirana kostima. Jedan bi mogao predstavljati jedan korak, a drugi bi mogao predstavljati drugi. Idemo na neku drugu tipku isprogramirati animirano kretanje pande. Za početak se želimo pomaknuti za 10 koraka, onda ćemo promijeniti izgled pande i pomaknuti se za još 10 koraka te vratiti izgled pande na početni – vidimo da se ništa ne vidi kada isprobamo, to je zato što moramo dodati i naredbu *wait*. Sada možemo dodati i naredbu *repeat* kako bi naša panda napravila više koraka odjednom.

Sada ćemo pogledati Python kod. Klikom na gumb  otvara se kod (slika desno) koji potom komentiram s učenicima i povezujem s objektnim programiranjem.

Primjer 3: Uvođenje drugog lika

Sada možemo na primjer uvesti još jedan lik na ekran. Vidimo da njega posebno programiramo neovisno o prvom liku. Idemo sad još uvesti varijable u priču, stvorimo novu varijablu na primjer neki brojač. Vidimo da kada kreiramo varijablu se pojave neke funkcije, možemo sakriti ili pokazati varijablu na ekranu, također postaviti i mijenjati vrijednost varijable. Recimo da želimo da naš novi lik izgovara vrijednost varijable, to možemo napraviti pomoću naredbe *say* i varijable koju ubacimo umjesto riječi unutar naredbe *say*. Možemo uvesti i naredbu *repeat* unutar koje ćemo u svakoj iteraciji prikazati vrijednost varijable, a potom je promijeniti za 1.



```
Python
1 from mblock import event
2 import time
3
4 @event.keypressed('r')
5 def on_keypressed():
6     sprite.say('bježim!')
7     for count in range(10):
8         time.sleep(0.01)
9         sprite.set_costume('costume2')
10        sprite.forward(10)
11        time.sleep(0.01)
12        sprite.set_costume('costume1')
13        sprite.forward(10)
14
15    sprite.say('BJEŽIM!', 0.1)
16
17
18 @event.keypressed('space')
19 def on_keypressed1():
20     for count2 in range(10):
21         time.sleep(0.1)
22         sprite.set_costume('costume2')
23         sprite.forward(10)
24         time.sleep(0.1)
25         sprite.set_costume('costume1')
26         sprite.forward(10)
27
28
29
30 @event.keypressed('left arrow')
31 def on_keypressed2():
32     sprite.forward(10)
33
34
35 @event.keypressed('right arrow')
36 def on_keypressed3():
37     sprite.forward(-10)
```

Sada prelazimo na zadatak za učenike.

Zadatak 1: Učenici programiraju pandu

Kada kliknemo na pandu, ona se mora okrenuti za 360 stupnjeva, odnosno oko svoje osi (hint: naredba *repeat* – očekujem da oni neće koristiti *repeat* pa ih želim navesti da želim vidjeti kako se okreće - animacija)

Rješenje:



Aktivnost 2: Grupni rad (60ak minuta)

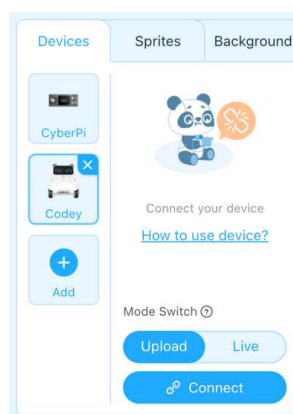
Podjela u grupe po 3-4 učenika (očekivano 3-4 grupe).

Kratko prezentiranje robota i njegovih mogućnosti.

- Ulaz: gumbi, kotačić (potenciometar), senzor zvuka i osvjetljenja, senzor orijentacije u prostoru (žiroskop; detektira nagib, trešnju i okretanje), prepoznavanje boja i prepreka, mjerač vremena, infracrveno svjetlo, bluetooth, wifi.
- Izlaz: LED ekran (matrica 8x16), zvučnik, RGB LED svjetlo, kotači.

Podjela robota svakoj grupi i spajanje robota na računalo.

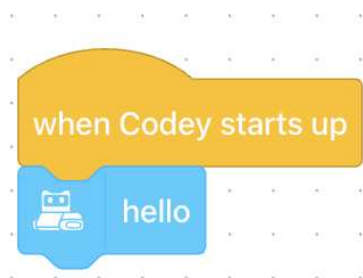
Sada ćemo pronaći našeg robota u *devices* (uređaji). Kliknemo na gumb *Add* i pronađemo Codeya, odaberemo ga i kliknemo OK. Potom robota moramo uključiti u računalo (USB ili po mogućnosti Bluetooth). Kliknemo na *Connect* i trebali bismo moći odabrati uređaj (OVDJE TREBA INSTALIRATI SOFTWARE mLink! Link: <https://mblock.makeblock.com/en-us/download/mlink/>).



Objasniti razliku između Live i Upload načina rada. Neke naredbe rade samo u jednom od ta dva načina, na primjer naredba „when Codey wakes up“ radi samo u Upload načinu, a naredba „when flag clicked“ radi samo u Live načinu rada. Kada smo u Upload načinu rada, program se prenosi na uređaj i tamo se ne može mijenjati. Kad želimo promijeniti program, to učinimo na računalo i ponovo prenesemo program na uređaj. U Live načinu rada program možemo mijenjati i odmah testirati promjene bez prenošenja. Također možemo kontrolirati našeg robota preko tipkovnice ili povezati likove na ekranu s robotom.

Primjer 4: Jednostavan primjer

Isprogramirat ćemo da kada uključimo Codeya, on nas pozdravi. Programirat ćemo u upload načinu rada. Koristit ćemo jednu od unaprijed definiranih radnji „hello“.



Kod možemo vidjeti s lijeve strane. Učenici će također trebati napraviti ovaj program, prebaciti ga na svoje robote i isprobati.

Primijetimo da kad prebacimo na live način rada, ikona „when Codey starts up“ postaje siva, što znači da tu naredbu ne možemo koristiti u tom načinu rada.

Sad prelazimo na zadatke za učenike.

Zadatci će biti podijeljeni u 2 listića. Svaka grupa prvo rješava zadatke s prvog listića, a kad ih riješe mogu prijeći na drugi listić. Prva tri zadatka su na prvom listiću, a druga tri su na drugom.

LISTIĆI:

1. Isprogramirajte *Codeya* tako da se klikom na tipke A, B i C prikažu pripadna slova na ekranu.
2. Isprogramirajte *Codeya* da kad ga protresete na ekranu prikaže animaciju otkucaja srca.
3. Isprogramirajte *Codeya* da prepoznaje neke boje. Na svaku prepoznatu boju postavite da RGB LED svjetlo svijetli u toj boji.

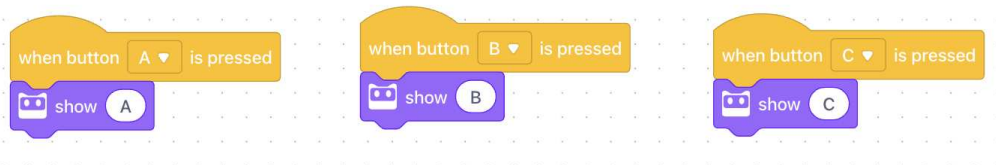
U sljedećim zadacima je bitno koristiti Live način rada i *broadcast* naredbu.

4. Isprogramirajte *Codeya* da se kreće pomoću tipki na tipkovnici.
 - a. Dodatno, dodajte četiri *spritea* (lika) na ekran i isprogramirajte mogućnost da se *Codey* može kretati i klikovima na likove po ekranu.
5. Isprogramirajte *Codeya* da se na pritisak tipki A i B lik na ekranu pomiče u lijevo i desno.
 - a. Dodatno, isprogramirajte na tipku C nešto po želji.
6. Isprogramirajte *Codeya* da kada padne mrak, on zijevne i ode spavati, a kada ponovo bude dan, on se probudi.
 - a. Dodatno, isprogramirajte da se pozadina na ekranu (*background*) promijeni u tamnu kada je mrak, a u svijetlu kada je dan.

Rješenja zadataka:

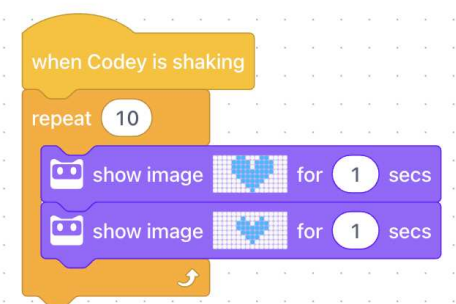
1. Isprogramirajte Codeya tako da se klikom na tipke A, B i C prikažu pripadna slova na ekranu.

Poveznica na rješenje: <https://ide.mblock.cc/#/?cloudProjectId=3458662>



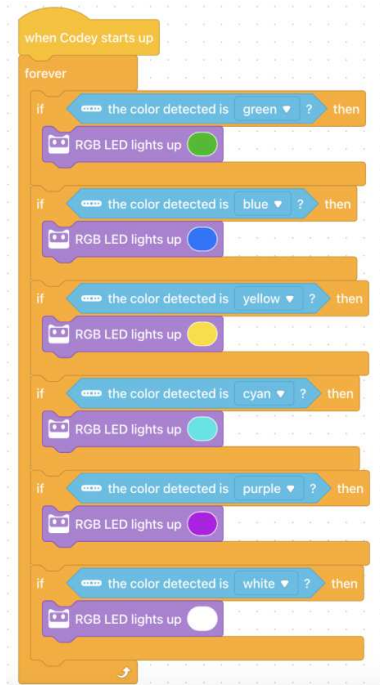
2. Isprogramirajte Codeya da kad ga protresete na ekranu prikaže animaciju otkucaja srca.

Poveznica na rješenje: <https://ide.mblock.cc/#/?cloudProjectId=2564017>



3. Isprogramirajte Codeya da prepoznaje boje, na svaku prepoznatu boju postavite da RGB LED svjetlo svijetli u toj boji.

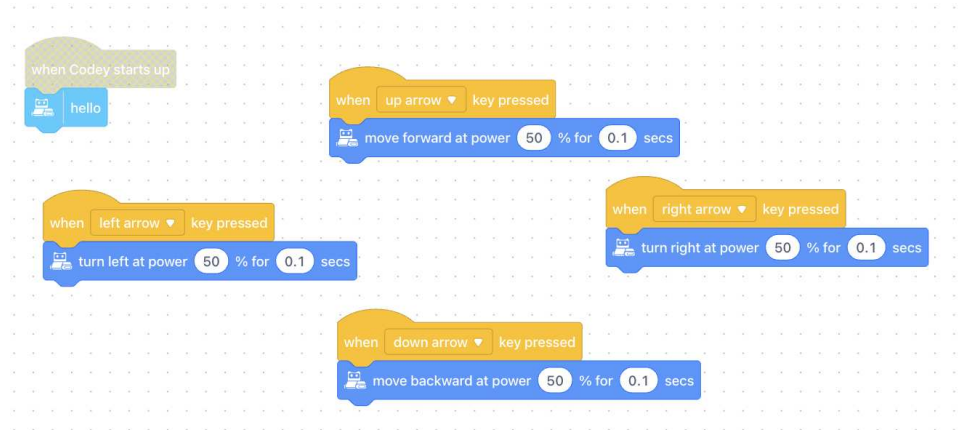
Poveznica na rješenje: <https://ide.mblock.cc/#/?cloudProjectId=3458682>



4. Isprogramirajte Codeya da se kreće pomoću tipki na tipkovnici (Live način rada)
- Dodatno, dodajte četiti spritea (lika) na ekran i isprogramirajte mogućnost da se Codey može kretati i klikovima na likove po ekranu (koristiti broadcast naredbu)

Poveznica na rješenje: <https://ide.mblock.cc/#/?cloudProjectId=3458689>

Rješenje zadatka:

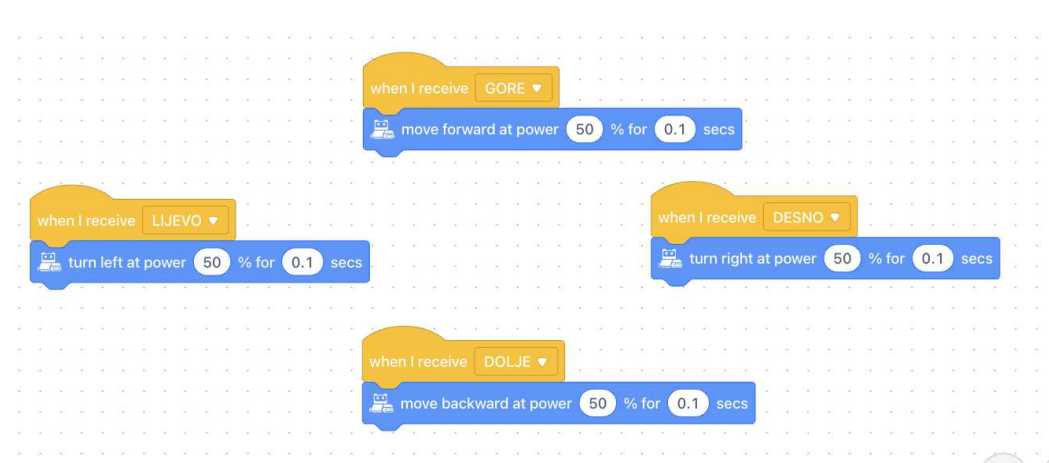


Rješenje pod zadatka:

Sljedeće funkcije su u svakom liku zasebno:



Sljedeće funkcije su za Codeya:



5. Isprogramirajte Codeya da se na pritisak tipki A i B lik na ekranu pomiče u lijevo i desno.
a. Dodatno, isprogramirajte tipku C po želji nešto vezano uz lik na ekranu

Poveznica na rješenje: <https://ide.mblock.cc/#/?cloudProjectId=3458694>



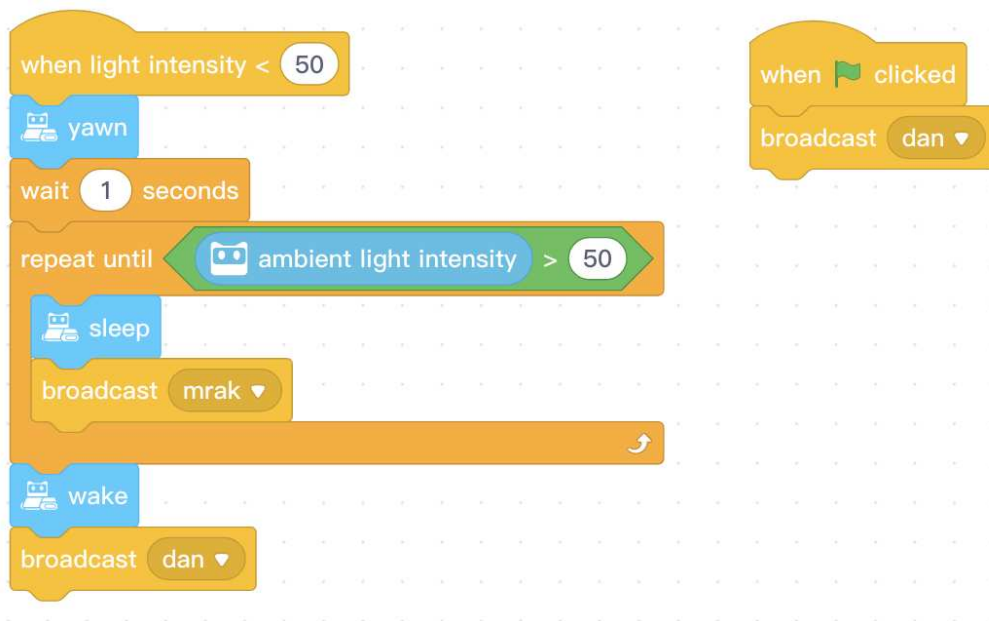
Codey:



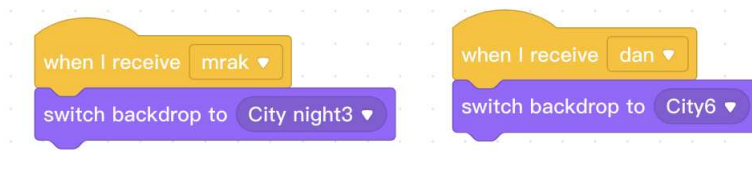
Panda:

6. Isprogramirajte Codeya da kada padne mrak, on zijevne i ode spavati, a kada ponovo bude dan on se probudi.
a. Dodatno, isprogramirajte da pozadina na ekranu (background) na početku bude svijetla pa se promijeni u tamnu kada je mrak, a u svijetlu kada je dan (koristiti broadcast naredbu i Live način)

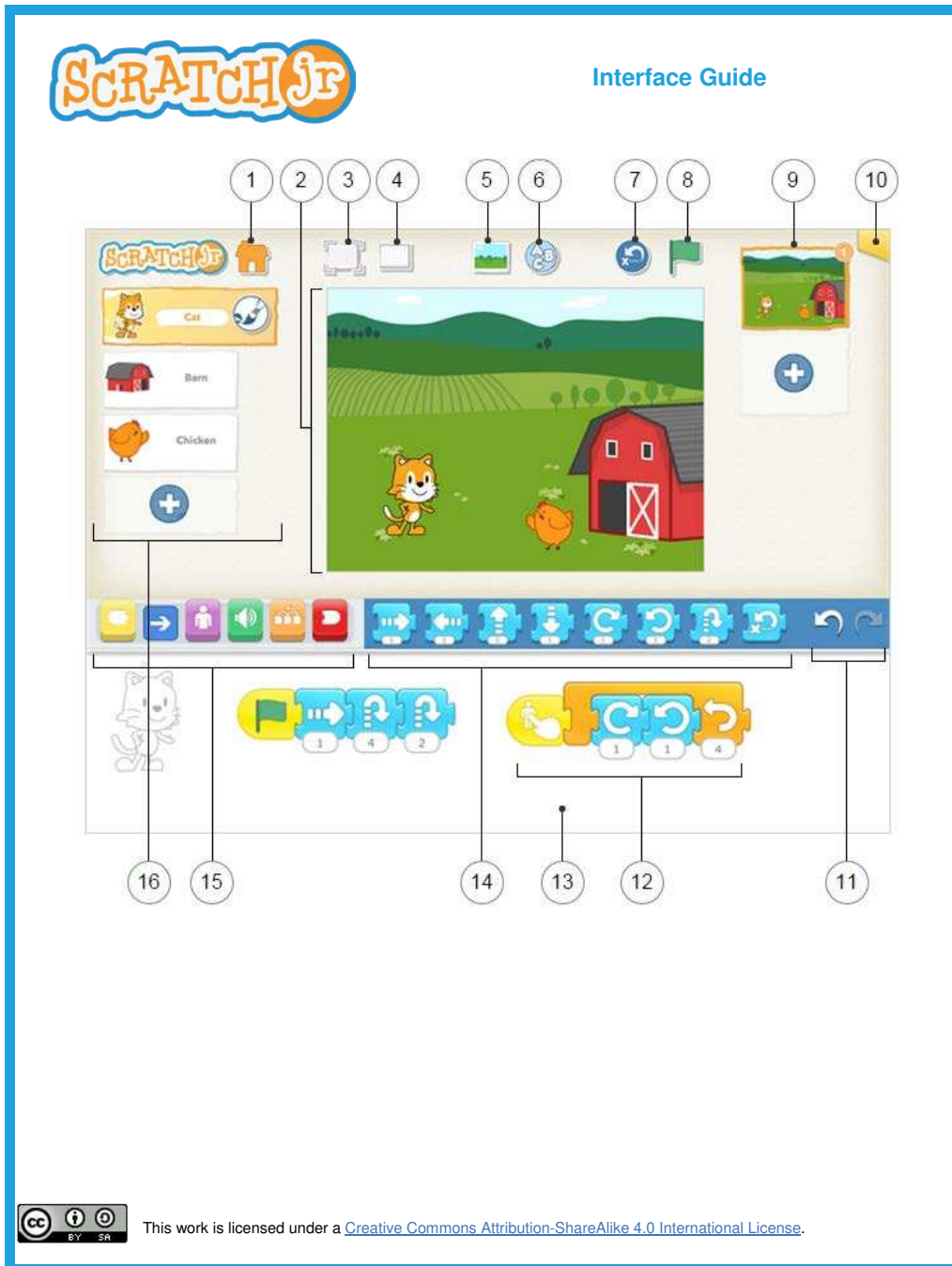
Poveznica na rješenje: <https://ide.mblock.cc/#/?cloudProjectId=3458699>




Funkcije pozadine (background):



4.2 Prilog 2





Interface Guide

- 1

Save

Save the current project and exit to the Home page.
- 2

Stage

This is where the action takes place in the project. To delete a character, press and hold it.
- 3

Presentation Mode

Expand the stage to the full screen.
- 4

Grid

Toggle on (and off) the x-y coordinate grid.
- 5

Change Background

Select or create a background image for the stage.
- 6

Add Text

Write titles and labels on the stage.
- 7

Reset Characters

Reset all characters to their starting positions on the stage. (Drag characters to set up new starting positions.)
- 8

Green Flag

Start all programming scripts that begin with a "Start on Green Flag" block by tapping here.
- 9

Pages

Select among the pages in your project -- or tap the plus sign to add a new page. Each page has its own set of characters and a background. To delete a page, press and hold it. To reorder pages, drag them to new positions.
- 10

Project Information


Change the title of the project and see when the project was created.
- 11

Undo and Redo

If you make a mistake, tap Undo to go back in time, reversing the last action. Tap Redo to reverse the last Undo.
- 12

Programming Script

Snap blocks together to make a programming script, telling the character what to do. Tap anywhere on a script to make it run. To delete a block or script, drag it outside the programming area. To copy a block or script from one character to another, drag it onto the character's thumbnail.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).



Interface Guide

- 13 **Programming Area**
This is where you connect programming blocks to create scripts
- 14 **Blocks Palette**
This is the menu of programming blocks. Drag a block into the programming area, then tap on it to see what it does.
- 15 **Block Categories**
This is where you can select a category of programming blocks:
Triggering Blocks (Yellow), Motion (Blue), Looks (Purple), Sounds (Green), Control (Orange), End Blocks (Red).
- 16 **Characters**
Select among the characters in your project -- or tap the plus sign to add a new one. Once a character is selected, you can edit its scripts, tap its name to rename it, or tap the paintbrush to edit its image. To delete a character, press and hold it. To copy a character to another page, drag it to the page thumbnail.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Bibliografija

- [1] Alnajjar, F., et al., (2021.), *Robots in Education: An Introduction to High-Tech Social Agents, Intelligent Tutors, and Curricular Tools*, New York: Routledge.
- [2] *Arduino Education Kits*, <https://www.arduino.cc/education/kits/>, (pristupljeno: 9.2.2024.).
- [3] Hrvatski robotički savez, *Novosti*, dostupno na: <https://hrobos.hr/category/novosti/>, (pristupljeno: 11.2.2024.).
- [4] *Kurikulum nastavnog predmeta Informatika za osnovne i srednje škole*, Narodne novine, izdanje: NN 22/2018, dostupno na: https://narodne-novine.nn.hr/clanci/sluzbeni/2018_03_22_436.html, (pristupljeno: 9.2.2024.).
- [5] *Logo History*, https://el.media.mit.edu/logo-foundation/what_is_logo/history.html, (pristupljeno: 5.2.2024.).
- [6] Manipulator, *Hrvatska enciklopedija, mrežno izdanje*, Leksikografski zavod „Miroslav Krleža”, dostupno na: <https://enciklopedija.hr/clanak/manipulator> (pristupljeno: 15.1.2024.).
- [7] Martin, F., et al., (2000), *To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines*, dostupno na: https://www.researchgate.net/publication/242467788_To_Mindstorms_and_Beyond_Evolution_of_a_Construction_Kit_for_Magical_Machines, (pristupljeno: 6.2.2024.).
- [8] Ministarstvo znanosti i obrazovanja, (2023), *Eksperimentalni kurikulum nastavnoga predmeta Informacijske i digitalne kompetencije za osnovne škole*, dostupno na: <https://mzo.gov.hr/UserDocsImages/dokumenti/Obrazovanje/OsnovneSkole/Eksperimentalni-kurikulum-nastavnog-predmeta-Informacijske-i-digitalne-kompetencije-za-osnovne-skole.pdf>, (pristupljeno: 9.2.2024.).

- [9] Molins-Ruano, P., Gonzalez-Sacristan, C., García-Saura, C., (2017). *Phogo: A low cost, free and “maker” revisit to Logo*, Computers in Human Behavior, dostupno na: https://www.researchgate.net/publication/320072960_Phogo_A_low_cost_free_and_maker_revisit_to_Logo, (pristupljeno: 5.2.2024.).
- [10] Nikolić, G., *Povijest automatskih uređaja – automata, preteče robota*, Povijest i filozofija tehnike, 8. simpozij PIFT 2019, str. 29.–48., dostupno na: <https://hro-cigre.hr/wp-content/uploads/2020/04/Nikolic-PIFT-2019-50.pdf> (pristupljeno: 15.1.2024.).
- [11] Nikolić, G., *Promjene značenja riječi robot u proteklih 100 godina*, Povijest i filozofija tehnike, 10. simpozij PIFT 2021, str. 13.–38., dostupno na: https://hro-cigre.hr/wp-content/uploads/2022/07/Filozofija_10simpozij_zbornik.pdf (pristupljeno: 15.1.2024.).
- [12] *Our Story*, <https://www.scratchfoundation.org/our-story>, (pristupljeno: 6.2.2024.)
- [13] Papert, S, (1980), *Mindstorms—Children, Computers and Powerful Ideas*, New York: Basic Books, Inc., dostupno na: <https://mindstorms.media.mit.edu/>, (pristupljeno: 5.2.2024.).
- [14] *Play to Learn Robotics and Coding*, <https://www.makeblock.com/pages/product>
- [15] Robot, *Hrvatska enciklopedija, mrežno izdanje*, Leksikografski zavod „Miroslav Krleža”, dostupno na: <https://enciklopedija.hr/clanak/53100> (pristupljeno: 15.1.2024.).
- [16] Sargent, R. et al., (1995.), *Building and Learning With Programmable Bricks*, e The Logo Foundation Newsletter, 3 (3), dostupno na: https://el.media.mit.edu/logo-foundation/resources/logoupdate/Logo_Update_v3n3.pdf, (pristupljeno: 6.2.2024.).
- [17] Watters, A., (2015) *Lego Mindstorms: A History of Educational Robots*, dostupno na: <https://hackeducation.com/2015/04/10/mindstorms> (pristupljeno: 6.2.2024.)

Sažetak

Tema kojom se bavi ovaj diplomski rad su roboti kao alati u nastavi Informatike s naglaskom na učenju programiranja. Rad se sastoji od tri glavna poglavlja. Prvo poglavlje proučava robote generalno, odnosno: njihovu povijest, (netrivijalnu) definiciju i elemente od kojih se sastoje. U drugom poglavlju se fokus prebacuje na robote u obrazovanju i njihovu povijest te se postepeno dolazi do robota koji se danas koriste u nastavi. Ti roboti su detaljno proučeni kao i njihove mogućnosti. Na kraju poglavlja su proučena neka popularna natjecanja u robotici te je prikazana tablica u kojoj su navedeni svi spominjani roboti, neke njihove karakteristike i cijena. Posljednje poglavlje proučava kurikulum te mjesto robota u odgojno-obrazovnim ishodima nastave Informatike. Prikazane su neke aktivnosti koje bi se mogle izvoditi u nastavi kao i priprema za blok sat s robotima koji je izveden u zagrebačkoj XV. gimnaziji u sklopu kolegija Metodička praksa iz informatike u srednjoj školi.

Summary

The subject of this thesis are robots as tools in the computer science education with an emphasis on learning programming. The thesis consists of three main parts. The first part studies robots in general, their history, definition, and components of which they consist. In the second part, the focus is shifted onto robots in education and their history, followed by gradually approaching the 21st century and practices in today's classrooms. The robots and their capabilities are studied in detail. At the end of the chapter, a table is created gathering all mentioned robots, some of their characteristics and price. Also, some popular robotic competitions are shown. The last chapter studies the curriculum and the place for robots in curriculum educational outcomes. Some activities and tasks which can be used in a classroom are shown.

Životopis

Rođena u Zagrebu 17. 12. 1995. godine. Pohađala sam osnovnu školu Mladost te zagrebačku XVIII. gimnaziju. Maturirala sam 2014. godine, a na Državnoj maturi iz Matematike sam točno riješila 95% ispita te time dosegla 281. mjesto u generaciji.

Nakon mature upisala sam preddiplomski sveučilišni studij Matematika na Prirodoslovno-matematičkom fakultetu u Zagrebu, a 2018. godine sam se prebacila na preddiplomski sveučilišni studij Matematika; smjer: nastavnički koji sam završila 2020. godine. Tada upisujem diplomski sveučilišni studij Matematika i informatika; smjer: nastavnički.