

Potočnica 1

Ana Lazarević

Ivana Baričević

## DECODING COMPUTERS

### PROGRAMMING

Djeca se dijele u parove, svaki par ima jednog programera i jednog robota. Programer zadaje niz jasnih uputa kojima pokušava navesti robota (čiji zadatak je pažljivo slušanje te pravilno interpretiranje uputa) van labirinta.

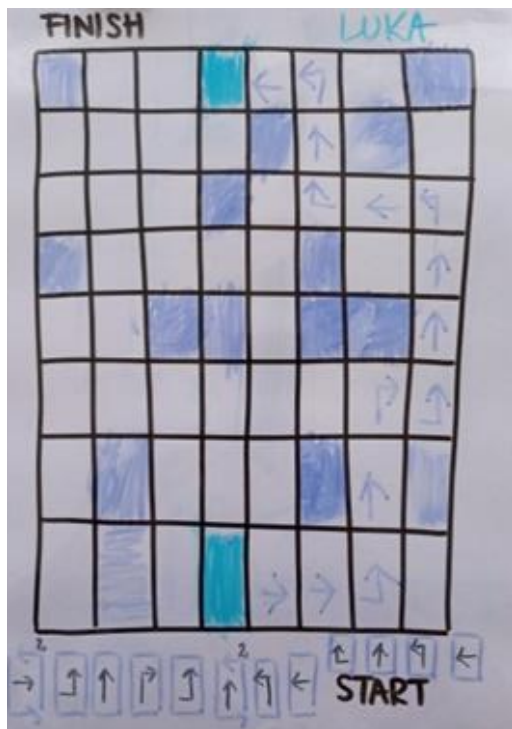


Šah- igra u kojoj gotovo svakodnevno uživa dio grupe, nadopunile smo različitim izazovima programiranja, kao i situacijskog rješavanja problema.

Djeca si međusobno postavljaju ploču na način da crni (ili bijeli) kralj treba doći do crne (ili bijele) kraljice, međutim, putem imaju na umu prepreke (ostale pijune) koje moraju imati na umu prilikom planiranja puta i programiranja kretanja kralja.



Također, igra ide i u obrnutom smjeru, pa isprogramirano kretanje kralja, djeca prikazuju pijunima na šahovskoj ploči.



Ili pokušavajući pronaći sve paralelne putanje za koje se može pripremiti algoritam kretanja kralja.



Djeca kreiraju igru pomoću prepreka napravljenih lego kockama.



Prepreke numeriramo, zatim opisujemo na koji način funkcionira i kako bi ju se moglo savladati. Isto bilježimo.

1. ZAMKA- trebaš skočiti preko nje

2. VATRA I LED- trebaš pritisnuti botun i izgasiti sistem da bi mogao proći

3. PIZZA- moraš pojest pizzu da ne se ne bi poskliznuo i pao

4. ŠLAG I KROKODIL- išao si gricnut šlag; krokodil te skoro pojeo dok si jeo šlag-moraš se boriti s krokodilom

5. NEVIDLJIVI ZID- pokušao si proći, ali si zapeo jer nisi vidio zid, ali, nakon što si udario, zid se aktivirao i sad ga vidiš-trebaš ga šutnuti nogom

6. OTROVNA GLJIVA- ako ju pojedeš, letiš u svemir-moraš ju preskočiti

7. TNT KLUPA- želiš leći kako bi odmorio, ako legneš eksplodirati će- moraš ju podignuti i baciti

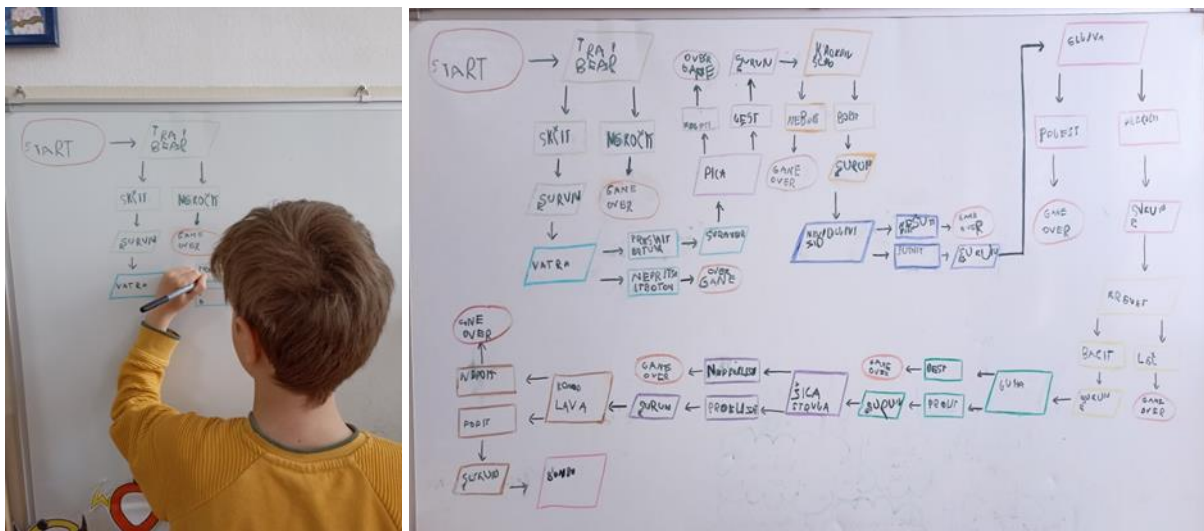
....

23. DVORAC- u kutiji s blagom u dvorcu si pronašao nagradu jer si prešao igru

## FLOW DIAGRAM

Na osnovi zabilješki stvaramo i dijagram toka-simbolički algoritam. Upoznajemo se s 4 glavna grafička simbola koja se koriste prilikom izrade dijagrama:

- izduženim krugovima koji označavaju početak ili kraj postupka
- pravokutnikom koji pokazuju upute ili radnje
- dijamanatom koji ističe gdje se mora donijeti odluka
- paralelogramom koji pokazuje ulaz i izlaz



## ALGORITHM

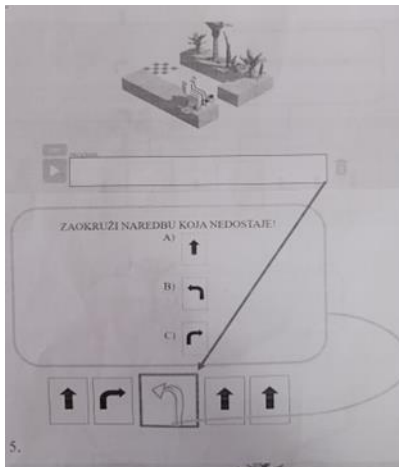
Djeca osmišljavaju jednostavnije igre u manjim grupama (najčešće po dvoje). Jedno dijete predstavlja programera, drugo testera (osoba koja je odgovorna za testiranje softverskih aplikacija kako bi se osigurala njihova funkcionalnost, kvaliteta i ispravnost-testeri rade na identifikaciji grešaka, provjeravaju ispravnost programa i provode testiranje kako bi osigurali da softver radi onako kako je namijenjen). Program stvara algoritam, kojeg zatim tester testira, pronalazi bug-ove (termin koji se koristi u programiranju i softverskom inženjeringu kako bi se opisala greška, nedostatak ili neispravnost u softverskom sustavu) kojima zajedno pokušavaju naći rješenje.





## IF LOOP

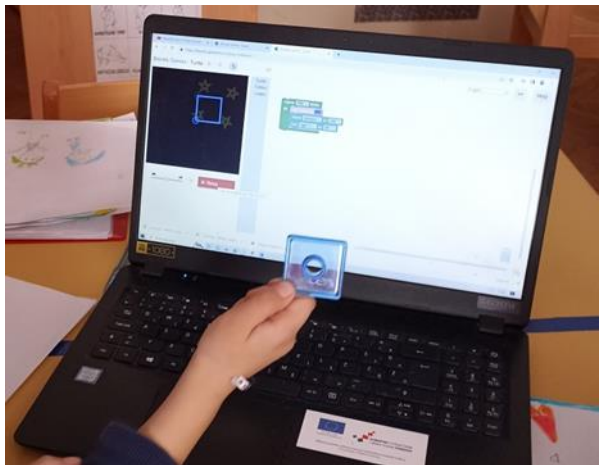
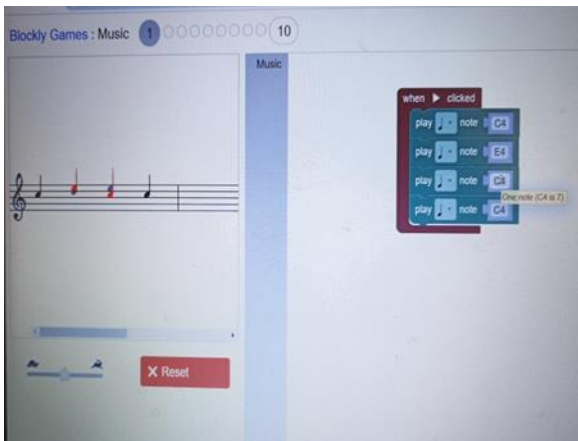
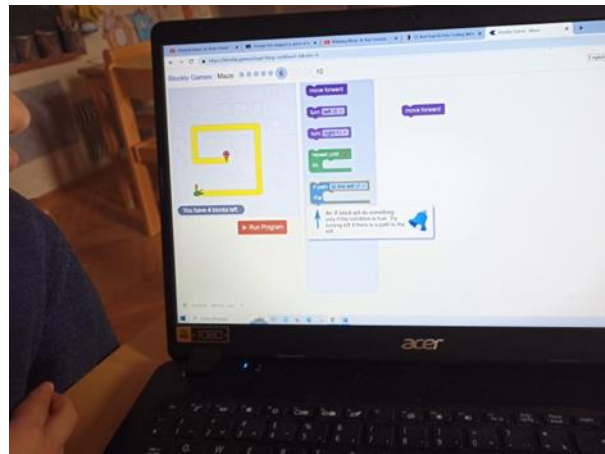
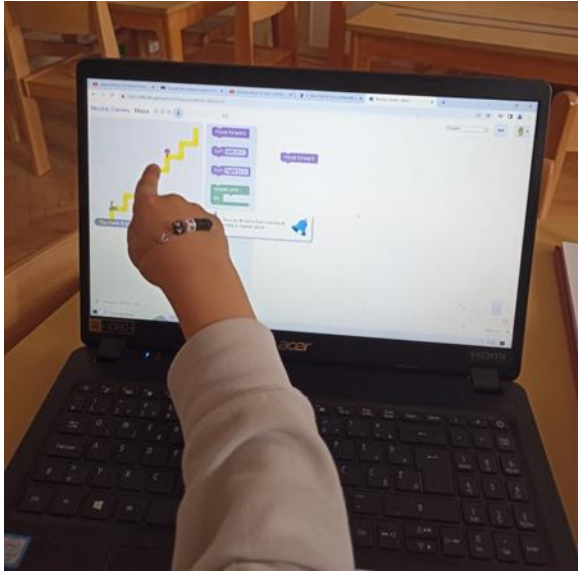
Kroz listiće djeca su imala priliku susresti se i sa sumiranjem naredbi te pojmovima poput petlji (loop). Petlja u programiranju omogućuje izvršavanje skupa naredbi više puta. To je kontrolna struktura koja smanjuje potrebu za ponavljanjem istih naredbi ručno.



## PROGRAMMING LANGUAGE

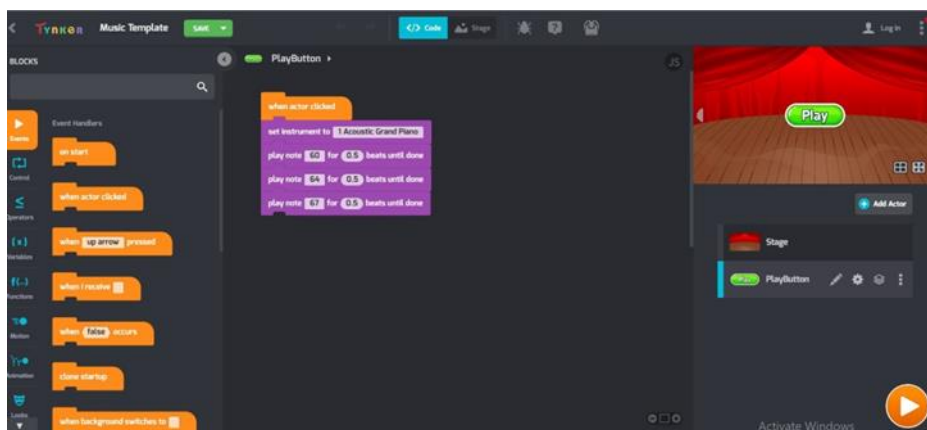
Programski jezik je formalni jezik koji se koristi za pisanje računalnih programa. Postoje mnogi programski jezici, svaki s vlastitim sintaksom, pravilima i funkcionalnostima.

Neki koje smo mi koristili u radu su ScratchJr i Blockly.



## TYNKER

Igru nadopunjujemo Tynker-om- obrazovnom platformom i alatom za učenje programiranja. Tynker koristi vizualno programiranje temeljeno na blokovima kako bi olakšao djeci razumijevanje osnovnih programerskih koncepta. Djeca koriste blokove za izgradnju logike programa.



## BINARY CODE

Istraživali smo kako funkcioniraju binarni brojevi. Na koji način se isključuju i uključuju bitovi kako bi prikazivali i pohranjivali informacije u digitalnom obliku.

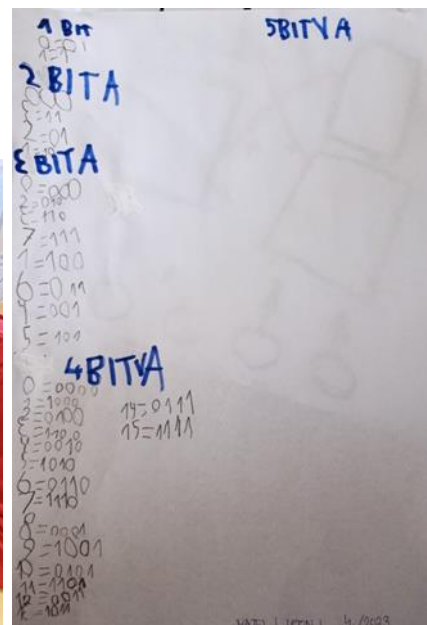




Brojeve binarnog sustava smo 'pretvarali' u ekvivalentnu vrijednost dekadskog sustava.



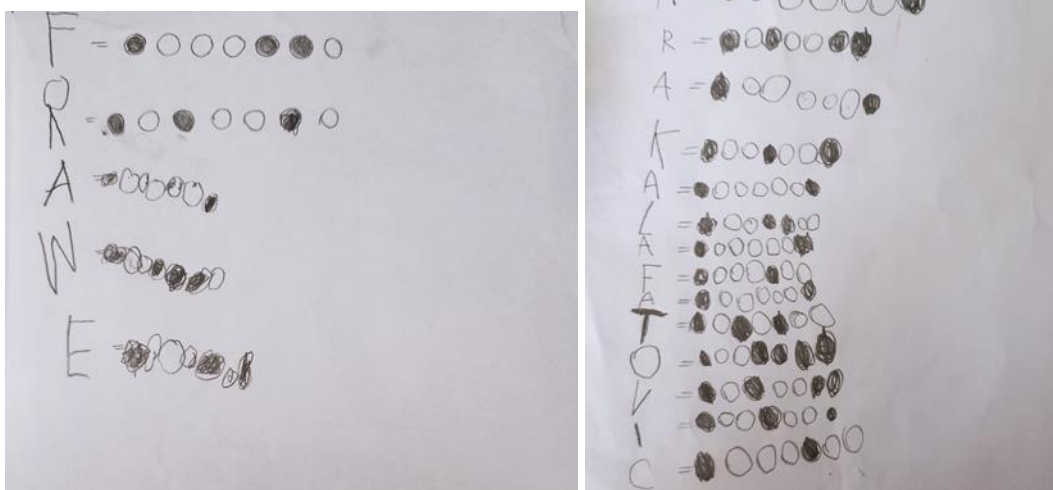
Pokušali smo saznati koliko opcija imamo ovisno o broju bitova s kojima raspolažemo.



Zatim smo numerirali slova abecede i istom logikom pretvarali binarne brojeve u latinična slova.



Varijantu uključenog bita reprezentirali smo bijelom bojom, isključenog crnom. Pa smo (koristeći se perlama istih boja) nizali vlastita imena ili imena članova obitelji, prijatelja...









## BUBBLE SORT

Bubble sort smo, u početku, koristili kao poligon (pokretnu igru) u kojoj je šestero djece predstavljalo šest brojeva ili slova (ulazne informacije) koji su ulazili u algoritam (poligon) i uspoređujući se (veći/manji) i razmjenjujući, ponavljali proces sve dok se ne sortiraju (od najmanjeg do najvećeg).



ulazni podaci



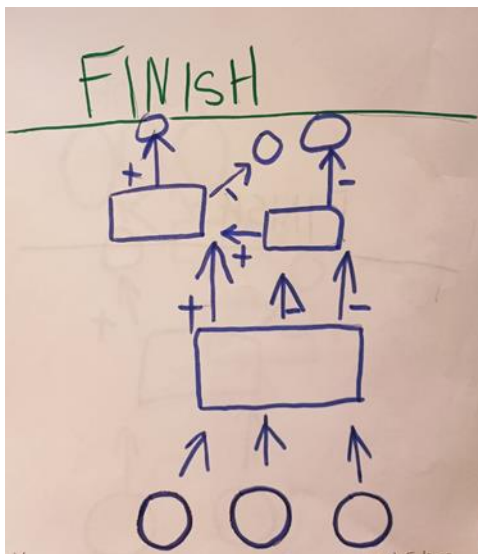
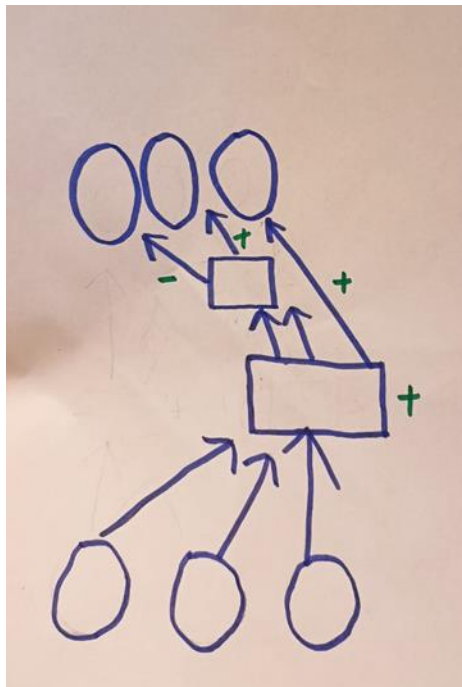
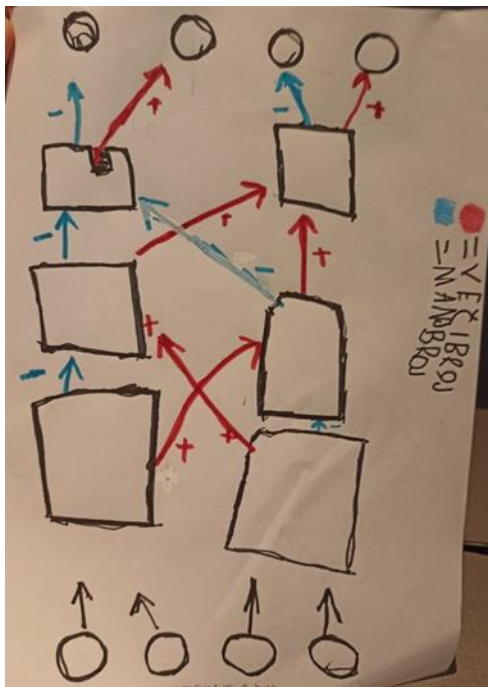
algoritam



sortirani podaci



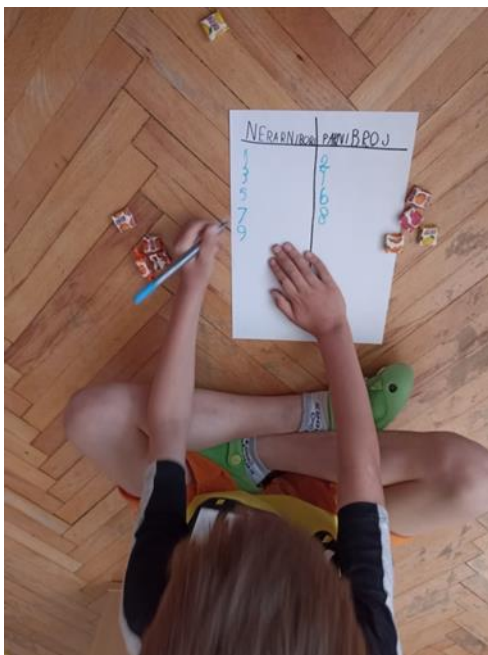
Nekoliko dana poslije djeca su samostalno (bez predloška) pokušavala osmisliti vlastite algoritme kojima bi sortirali podatke. Izazov su bili algoritmi za dva, tri i četiri podatka.



## PARITY MAGIC

U informatici se paritet često koristi u kontekstu otkrivanja pogrešaka. Bitovi parnosti dodaju se podacima kako bi se osigurala cjelovitost prenesenih informacija. Bit parnosti postavljen je na 0 ili 1, ovisno o tome je li broj bitova u podacima (obično binarnom) paran ili neparan. Provjerom bita parnosti na prijemnom kraju mogu se otkriti greške u prijenosu. Ako primljeni podaci imaju neispravan paritetni bit, to znači da je došlo do pogreške tijekom prijenosa.

Pomoću bombona otkrivali smo koji brojevi (do 10) su parni (oni koje možemo podijeliti na dva djeteta bez otvaranja i prepolavnjanja bombona), a koji neparni.



Zatim smo to znanje iskoristili kako bi pronašli greške u bitovima

