

# Scratch - programiranje na zabavan način

Mirena Maljković, učiteljica informatike<sup>1</sup>, prof.dr.sc. Mario Dumančić<sup>2</sup>  
OŠ Rapska i OŠ Jure Kaštelana Zagreb, Učiteljski fakultet Sveučilišta u Zagrebu  
[mirena.maljkovic@skole.hr](mailto:mirena.maljkovic@skole.hr), [mario.dumancic@ufzg.hr](mailto:mario.dumancic@ufzg.hr)

## Sažetak

*Programiranje je način razmišljanja. Cilj uvođenja programiranja u osnovnoj školi od prvog razreda tj. najranije dobi nije stvoriti isključivo buduće IT-ovce odnosno programere. No važno je djecu podučavati problemskom razmišljanju odnosno vještinama rješavanja problema. Scratch je zajednica koja svim učenicima približava programiranje zbog jednostavnosti sučelja. Također izaziva pozitivne emocije toliko važne za uspjeh procesa učenja zbog atraktivnosti koju nudi mogućnošću biranja likova, pozadina, zvukova i ostalim elementima. Učenici petih razreda stoga su s lakoćom ovladali početne korake programiranja te s velikom intrinzičnom motivacijom pristupili izradi vlastitih računalnih igara i ostalih aplikacija. Scratch je sučelje koje igrifikacijom učenicima približava ono što je većini odbojno te time povećava mogućnost ovladavanja tako važne vještine problemskog načina razmišljanja.*

## 1. UVOD

Novi kurikulum predviđa izbornu nastavu informatike od prvog razreda osnovne škole [1]. Unutar njega predviđeno je usvajanje znanja programiranja i algoritamskog razmišljanja. A zašto je ono važno? Generacije koje će stupiti na tržište rada više ne smijemo obrazovati po principu da ih pripremamo u školi određenim znanjima za određeno zanimanje. Zanimanja kojima će se nove generacije baviti još i ne postoje. Vrijeme je da škola priprema djecu za novo tržište rada. Nije više važno u učenike “uliti” mnoga znanja već ih moramo naučiti razmišljanju, rješavaju novonastalih problema te naučiti ih učiti ono što im je u datom trenutku potrebno. Prensky [2] je osmislio *Plan B* po kojem učenici uče dok rade tj. učimo ih novim vještinama, ali glavna je vještina da učenici radeći konkretne zadatke stječu nova znanja.

I upravo na taj način potrebno je postaviti učenje programiranja u osnovnoj školi. Ne možemo očekivati da će svi učenici postati programeri, ali uvođenjem programiranja mi im omogućavamo usvajanje vještina problemskog i algoritamskog razmišljanja. O programiranju se do nedavno smatralo kao aktivnostima za “odabrane učenike”, “matematičare” koji su posjedovali intrinzičnu motivaciju, a razlog tome možda leži u činjenici da su programski jezici (editori za pisanje koda, razvojna sučelja i sl.) novijim generacijama djelovali neprivlačno jer

su učenici naučeni na boje, bogata grafička sučelja, moderna grafička rješenja za razliku od uobičajenog pisanja naredbi kroz naredbeni redak.

Osvještavanjem važnosti činjenice koliko je programiranje važno od najranije dobi kontinuirano se osmišljavaju novi programski jezici prihvatljivi za rano učenje programiranja. Jedan od takvih modernih pristupa je razvojna okolina **Scratch**. Svojim izgledom te lakoćom pristupa i kreiranja koda privlači učenike koji do sada prema programiranju ne pokazuju nikakve simpatije. Scratch jednostavnim setom naredbi kreiranih tako da podsjećaju na igru slaganja “puzli” pruža široki spektar mogućnosti kojim se mogu kreirati složena programska rješenja, te ovladati problemskim i algoritamskim načinom razmišljanja.

U samom početku metodički je važno učenicima prikazati programiranje kao osobni ili grupni projekt koji može biti interaktivna priča, računalna igra ili nešto drugo. Svaki projekt sastoji se od niza manjih zasebnih aktivnosti, aktivnosti se sastoje od niza procedura, a procedure od niza naredbi. Stoga osnovni dio uvoda u programiranje u osnovnoj školi je postaviti projektni problem, te odrediti cilj projekta. Što želimo kreirati, što time želimo postići? Primjerice računalnu igru. Koje funkcionalnosti želimo postići? Koje likove želimo kreirati? Kakav će biti tijek računalne igre? itd.

Kad se postavi projekt potrebno je s učenicima definirati procedure, likove, njihove mogućnosti, pri čemu je važno razvijati algoritamsko razmišljanje kod učenika. Učenicima se pojam algoritma i algoritamskog razmišljanja mora približiti od njima poznatih elemenata koji su im svakodnevno dostupni. Primjerice mama želi napraviti tortu za rođendan. To je njen cilj. Kako će je napraviti? Prvo joj je potreban recept kako bi uopće mogla odabrati sastojke. Recept je algoritam. Kroz njega određujemo koji sastojci su nam potrebni i na koji ih način obrađujemo kako bi napravili tortu. Kroz ovaj primjer učenici uviđaju elemente i postupke s istima kako bi došli do cilja. Sastojci i postupci njima su elementi koje mogu pronaći u razvojnom sučelju (programske naredbe). Svaki programski jezik ima kreiran set programskih naredbi.

## 2. RAZVOJNI ALAT SCRATCH

Scratch je programski jezik i programsko okruženje koji omogućava jednostavno stvaranje interaktivnih priča, igara i animacija, te dijeljenje radova preko računalne mreže. Programsko sučelje učitava se unutar Web preglednika stoga je osnovni uvjet pristup mreži. Samo programsko okruženje ne zahtijeva nikakve specifične dodatne računalne resurse. Scratch

je projekt od Lifelong Kindergarten Group pri MIT Media Lab [3]. Nakon kreiranja programskog jezika i programskog okruženja MIT Media Lab je dodijelio licence slobodnog koda omogućavajući slobodno dijeljenje i korištenje. Danas je aktivno oko 12.297.156 online korisnika što predstavlja nevjerojatnu brojku. Scratch je preveden na 40 jezika te broj korisnika i njihovih projekata raste iz dana u dan [4]. Posebno je važno što se razvijaju zajednice Scratch programera pružajući jedni drugima potporu u realizaciji projekata. Svi projekti mogu biti javno dostupni pružajući drugima mogućnost učenja kroz iste ili dalje zajedničko razvijanje zanimljivih projekata.

## **Scratch u školi**

Scratch zbog svog pristupa programiranju i jednostavnošću programske razvojne okoline postaje zanimljivo rješenje za primjenu u primarnom obrazovanju s ciljem učenja programiranja i razvoju algoritamskog razmišljanja. Škole u SAD-u i Engleskoj prve su započele uvođenjem Scratcha u nastavni proces. Vrlo brzo su i ostale europske zemlje započele s pripremama uvođenja Scratcha u primarno obrazovanje. U Republici Hrvatskoj u primarnom obrazovanju koristi se Logo kao programski jezik. Smatramo da uz Logo ima dovoljno mjesta za novi programski jezik. Scratch ne zahtijeva posebnu infrastrukturu stoga se može započeti i s uvođenjem Scratcha u naše primarno obrazovanje [5]. Autori ovog rada željeli su jednim jednostavnim studijom slučaja (case study) otvoriti vrata Scratch programskom okruženju unutar primarnog obrazovanja. U OŠ. Jure Kaštelana u 5. razredu s učenicima smo započeli rad unutar novog programskog okruženja. Kroz postojeći nastavni plan i program prilagodili smo nastavu u novom okruženju. Nakon završene studije pitali smo učenike za njihovo mišljenje (kroz kratki anketni upitnik), te analizirali njihove uradke.

### ***2.1 Prvi koraci u Scratch-u***

Nakon što se učenicima pojasni osnovne pojmove programiranja i važnost njihovog svladavanja, započeto je upoznavanje sa Scratch razvojnim sučeljem. Učenici učitavaju Scratch sa stranice <https://scratch.mit.edu/>.

Pokaže im se reklamni video u kojem se kratko opisuju mogućnosti Scratcha (slika 1).



**Slika 1: Reklamni video na službenom web sjedištu Scratch zajednice**

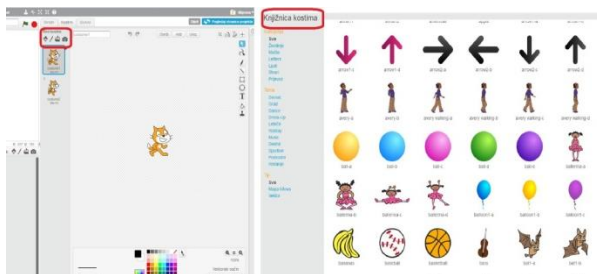
Slijedi pristup zajednici. Kako bi se omogućio pristup zajednici učenici moraju stvoriti svoj profil na Scratchu kroz pet koraka. Zatim moraju potvrditi pristup klikom na hipervezu na elektroničkoj pošti.

Nakon što učenici pristupe zajednici pokazuju im se osnovne mogućnosti Scratcha tako da odaberu opciju za kreiranje vlastitog uratka (slika 2).



**Slika 2: Tipka za početak izrade vlastite igrice**

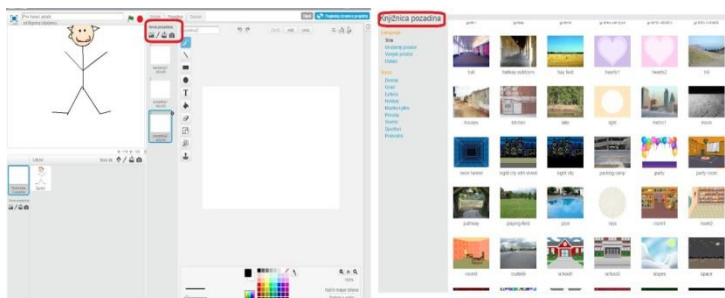
**Likovi (Sprite):** Prvo se upoznaju s likovima. Lik u Scratch-u zove se **Sprite**. Početni je mačak, ali u bazi ih ima puno više (Slika 3).



**Slika 3: Likovi (Sprite-ovi) u Scratchu**

Osim što se može izabrati Sprite iz galerije, može se učitati kao slika iz datoteke, kao slika s kamere ili se može napraviti vlastiti lik. Učenicima se pokaže gdje i kako se može napraviti Sprite te ga onda oni sami izrađuju.

**Pozadine (Pozornice):** Slično kao i sa Spriteovima postoje i pozadine ili pozornice. One mogu biti dodane iz galerije pozadina, mogu se učitati s računala ili se mogu napraviti samostalno (Slika 4).



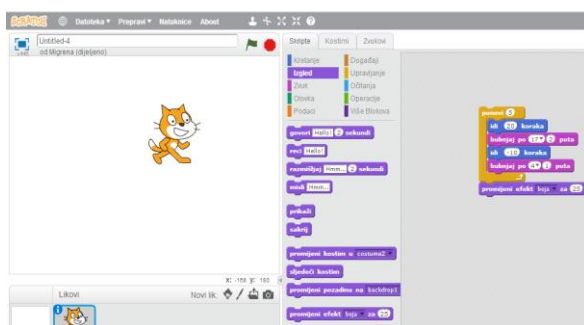
**Slika 4: Pozadine u Scratchu**

Učenicima se pokazuju sve opcije, a zatim im se zadaje zadatak da naprave svoju pozadinu.

**Naredbe (Blokovi naredbi):** Učenicima se objašnjavaju osnovne naredbe i kako ih pokrenuti. Počinje se s kretanjem. Klikne se na kategoriju Kretanje, odabere se naredba, primjerice *idi \_ koraka* (broj koraka se može mijenjati). Zatim se mogu dodavati ostale naredbe, primjerice iz kategorije Zvuk. Kad se mačak prošeće 25 koraka čuje se zvuk Bubnjeva. Zatim se mačak vrati za određen broj koraka i opet se čuje zvuk bubnjeva. Odmah se objašnjava vrlo važna kategorija Upravljanje. U njoj su naredbe *ponavljaј X puta*, *ponovi, ako - onda - inače* (*If-Then-Else*) i ostale koje se učenicima objašnjavaju tijekom rada.

Važno je učenicima objasniti kako se blokovi dodaju i vežu te kako se razdvajaju i uklanjaju.

U konkretnom “Prvom programčiću” naredbe se podese tako da se dobije dojam da mačak pleše jer se iste radnje ponove N puta i kad je sve gotovo, Mačak promijeni boju (kategorija Izgled) (Slika 5).



**Slika 5: Blokovi naredbi**

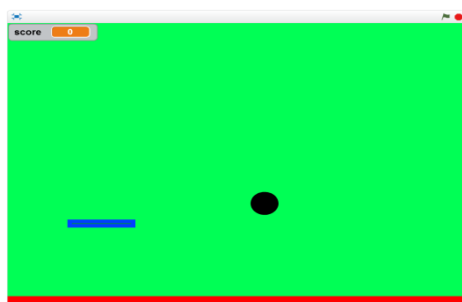
Zatim se objasni još jedan vrlo važna kategorija Događaji, jer se u njoj postavljaju osnovne kontrole za početak odvijanja programa, ali i ostali “triggeri” (okidači) kasnije u

programu. Za pokretanje igrice najbolje je upotrijebiti pokretač klikanja na zelenu zastavicu ili pritiska razmaknice na tipkovnici. Nakon što se postave neke osnovne naredbe učenici pokreću program. Pokazuje im se Pregled (Preview) sučelje u kojem uživaju u efektima svog rada.

Učenici spremaju svoje prve uratke i daje im se za zadatak da naprave svog lika i nekoliko osnovnih radnji s njim koji se ponavljaju pet puta.

## ***2.2 Postavljanje problema***

Nakon što se s učenicima provede osnovne radnje upoznavanje programskog okruženja u Scratch-u učenicima je postavljen programski problem koji je za ovu studiju slučaja izrada ljestovne računalne igre. Napominje im da i oni počnu razmišljati kakvu bi igricu htjeli napraviti. Pri predstavljanju programskog problema potiče se učenike na razmišljanje o svim elementima računalne igre. Učenike se pritom potiče na algoritamsko razmišljanje. Kako bi potaknuli razmišljanje programski zadatak je prikazan zorno učiteljica učenicima prezentira ranije računalnu igru, te ju istodobno učenici i pokušavaju igrati. (Slika 6).



**Slika 6: Prikaz igrice Pong**

Nakon što se učenici zabave, zadaje im se zadatak da napišu osnovne elemente u igrici. Likove, pozadine, što je uloga kojeg lika, koji je cilj igre.

Kreće se korak po korak u izradu programskog zadatka. Svaki korak započinje nizom pitanja za razmišljanje.

Pojedini blok se odvija u jednom vremenskom slijedu. Izdvojeni blokovi se mogu odvijati u paraleli. Svima je definiran isti pokretač, stiskanje razmaknice. Iz razloga što želimo ponovljena odvijanja svakog bloka, umjesto jednokratnog izvršenja, koristi se naredba *ponavljaj* koja vrti petlju bez kraja.

## ***2.3 Definiranje okoline i elemenata***

Učenici izrađuju Pozadinu i likove po točno zadanim uputama za računalnu igru Pong. Naglasak je na Crvenu liniju na dnu, zatim na veličinu loptice (ne smije biti prevelika ni premala) te Paddle mora odgovarati veličini loptice.

Zaključuje se da postoje tri vrste Pozadina u konkretnoj igrici - Početna, Sretan kraj i Gubljenje u igri. Crtaju se sve opcije. Crta se pozadina i oblici.

## ***2.4 Događaji u igrici***

**Igra započinje** pritiskom tipke - razmaknica. Loptica se kreće u raznim smjerovima, a kada dođe na rub okreće se. Učenicima se objašnjava podešavanje **kretanja po koordinatnom sustavu**. Zatim se uvodi blok ponavljanja kako loptica nikad ne bi nestala, a kad se dogodi događaj da *Loptica dođe na rub*, da se okrene i nastavi kretati.

### **Gubitak bodova:**

Kad *Loptica dotakne dno* što se događa? Gube se bodovi. Kako smo definirali dno? Crvenom crtom na pozadini. Uvodi se naredba *Ako* unutar koje se definiraju događaji. Dakle, ako *Loptica dodirne crvenu boju*, vrati se na sredinu ekrana, sviraj zvuk (povratna informacija igraču) i score (bodove) umanjí za jedan bod. Sve se ponavlja.

### **Dobivanje bodova:**

Kada loptica dotakne Paddle broj bodova se povećava za 1. Dodaje se zvuk *Laser* kako bi igrač osvijestio dobítak boda. Igra se nastavlja dalje, stoga se loptici daje naredba da se okrene za 180 stupnjeva, tj. nastavlja igru. Pritom se stavlja poček od 1 sekunde kako bi se loptica stigla udaljiti od Paddle-a i kako bi se igra mogla nastaviti.

### **Pobjeda u igri:**

S učenicima se raspravlja o tome što će se dogoditi kad se u igri pobijedi. Prvo zaključuju što je okidač (trigger) za pobjedu. Dolazi se do zaključka da određeni broj skupljenih bodova donosi pobjedu. Treba postaviti problem tj. petlju AKO – ONDA. Ako se postigne 30 bodova, igrač dobiva povratnu informaciju (feedback) o pobjedi. Učenici dolaze do rješenja da se za povratnu informaciju mora promijeniti okolina. Učenici sami izrađuju "*pobjednički ekran*". Dalje se dogovara tijek u igrici. Razmatra se o opcijama da se može napraviti prelazak na sljedeću razinu. Za početak, pobjeda znači kraj računalne igre. Učenicima se ostavlja na razmišljanje da kad krenu raditi svoju igricu što će osmisliti kao nastavak igre.

## Izgubljena igra:

Nakon što su osmislili tijek do pobjede, učenici razmišljaju o kraju u kojem igrač izgubi. Dolazi se do zaključka da je blok naredbi sličan, osim što je okidač u ovom slučaju score (broj bodova) koji padne na nulu, što pokreće daljnji tijek događaja u igrici tj. pojavljuje se "gubitnički ekran". Učenici sami osmišljavaju gubitnički ekran.

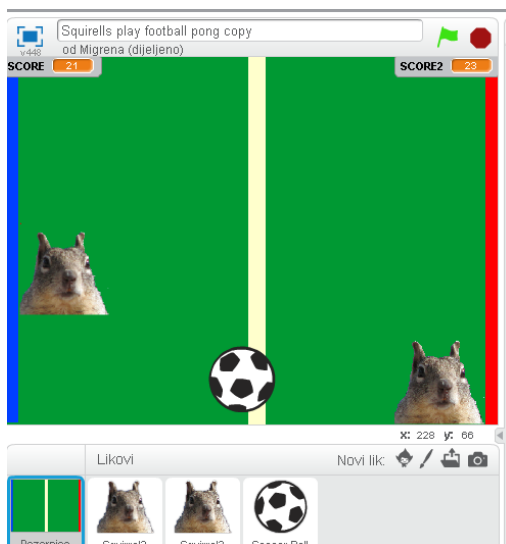
Nakon kreiranja, učenici sami isprobavaju svoje igrice.

## 2.5 Učenici izrađuju svoje igre

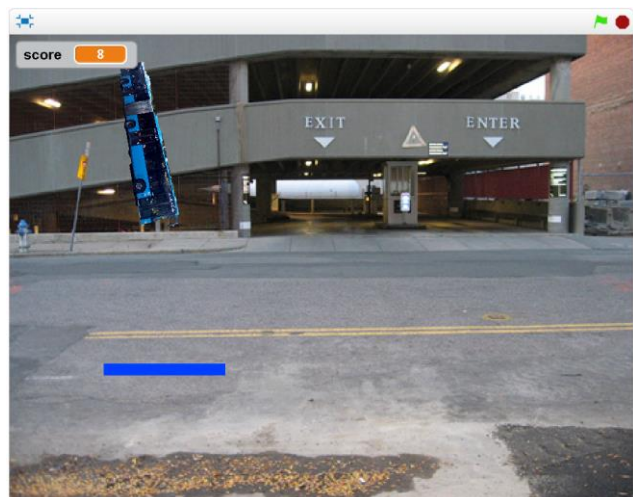
Nakon što su upoznali programiranje kao problemski način razmišljanja, naučili što je algoritam i upoznali se sa sučeljem u Scratch-u, učenici izrađuju svoje igre. Zadano im je da igru Pong modificiraju po vlastitim željama. Također, mogu napraviti svoju, novu igru od početka. U tom slučaju razrađuju ideju i pišu algoritam. Postavljaju problem i osmišljavaju rješenja. Koji su likovi u igrici? Koji je cilj igre? Kome je igrica namijenjena? Kako se skupljaju bodovi.

Primjeri računalnih igara koje su učenici osmislili i izradili samostalno:

### Vjeverice igraju nogomet

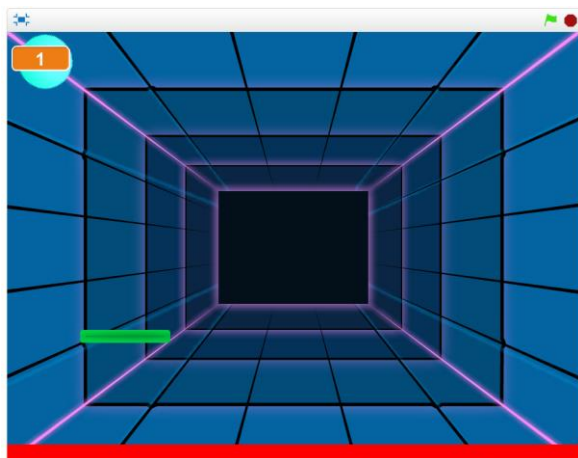


### Varijacija Pong-a (autobus koji se odbija od Paddle-a)

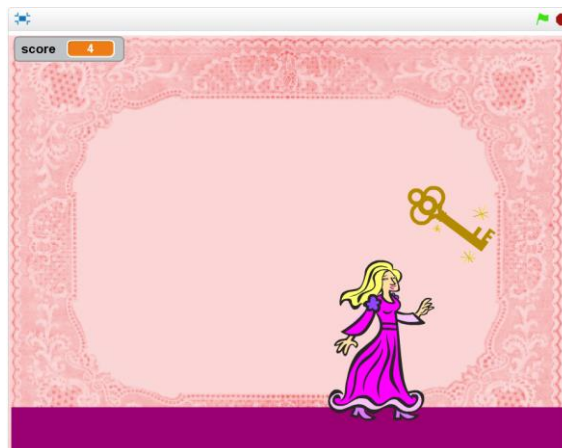




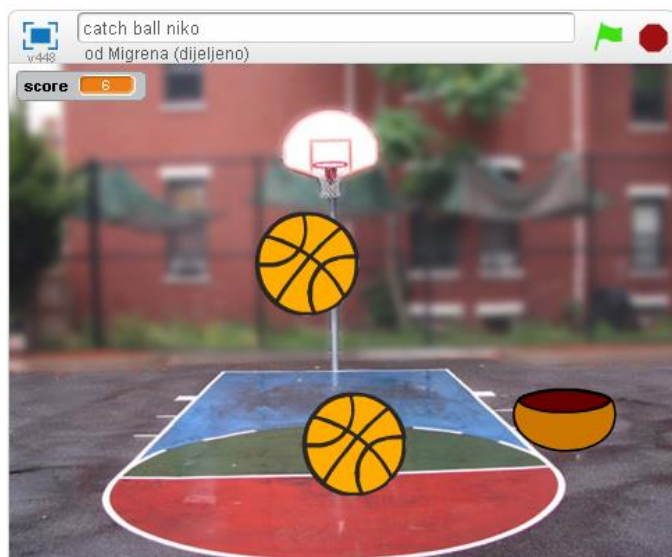
**Varijacija Ponga- loptica mijenja boju svaki put kad dotakne Paddle**



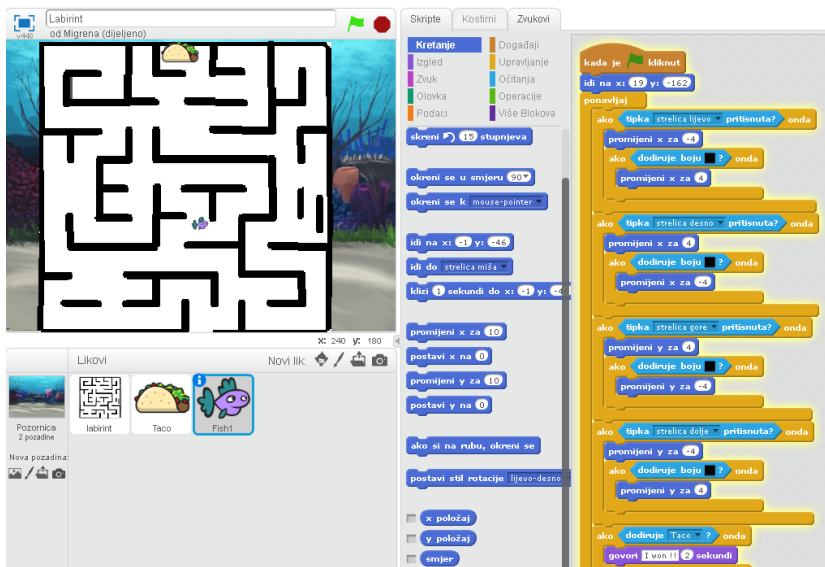
**Varijacija Pong-a – Princeza hvata ključ**



**Hvatanje lopti u košaru**



Učenici su sami istraživali i naučili kako napraviti Labirint. Neki primjeri:



### 3. ZAKLJUČAK

Samostalniji učenici napravili su vlastite igre. Postigao se najpoželjniji učinak, a taj je da su učenici sami osmislili igrice. Kada nisu znali riješiti problem tražili su informacije na internetu, a s obzirom da je Scratch zajednica popularna u svijetu, učenici su informacijama lako pristupili.

Dakle, učenici su sami nalazili informacije u svrhu rješavanja problema te su ih sami i rješavali. U vrijeme kad digitalni domoroci [9] nemaju, unatoč vjerovanjima, visoku razinu medijske pismenosti te internet koriste isključivo za zabavu, vrlo je važna činjenica da su učenici svoje vještine "surfanja" iskoristili za nešto korisno. Učenici koji nisu postigli potpunu samostalnost u novom okruženja Scratch-a, napravili su vlastite inačice Pong igre. Mijenjali su pozadine i likove. Time su svaki na svoj način ispričali priču. Također su nastale i verzije sa više kugli, različitim događajima u programu gdje primjerice kugla mijenja boju u dodiru sa Paddle-om.

Nakon što su učenici pokazali i zaigrali svoje igrice, provedena je anketa o dojmu učenja programiranja u Scratch-u.

Istraživanje opisano u ovom članku provedeno je na dva peta razreda u OŠ Jure Kaštelana u Zagrebu, a anketi je pristupilo 40 učenika. Pitanja koja su se odnosila na Scratch ukazuju da 63,9% učenika programiranje doživljava kao zabavno. Ovaj podatak ukazuje na specifičnosti programskog okruženja Scratch-u koje je prilagođeno dobi učenika. Ovo možemo potkrepiti i pitanjem koje se odnosilo na težinu programiranja gdje je samo 38% učenika iskazalo mišljenje da je programiranje teško, a 5% da je dosadno. Učenički komentari uglavnom su pozitivni: "Bilo je zabavno"; „Bilo je nešto drugačije, ali je bilo zabavno“; „Jako mi se svidjelo“; „Cool je“; „Sve ja jako dobro čak i zabavno“; „Na početku teško, a onda zabavno“; „Bilo je jako zabavno i kreativno“; „Svidjelo mi se raditi likove“; „Svidjelo mi se zato što možemo napraviti svoje igrice“; „Bilo je zabavno i puno sam toga naučio“; „Jako je zabavno i možeš biti kreativan!“; „Svidjelo mi se zato što se mogu raditi razne app-ovi“; „Vrlo edukativno i zabavno“; „Programiranje mi se jako svidjelo zato što je zabavno i zanimljivo“.

Scratch je zanimljivo i kreativno programsko okruženje koje bi trebalo više uključiti u nastavni proces primarnog obrazovanja. Nama autorima je neshvaljivo da još uvijek u školama možemo pronaći nastavu programiranja u Basicu koji je kvalitetan programski jezik ali je na žalost "ostario". Scratch se pokazao kao zanimljivo i poticajno programsko okruženje

koje ima potencijal za korištenje u primarnom obrazovanju. U slijedećim istraživanjima potrebno je potkrijepiti ovu studiju slučaja na većem uzorku učenika i sa kompleksnijim zadacima.

Zaključak možemo svesti na jednu rečenicu. Učenici vole stjecati nove i zanimljive vještine, oni vole učiti. Iznimno je važno omogućiti im to na način koji je njima prihvatljiv, a to je, naravno, kroz zabavu. Obzirom da postoji niz alata koji učiteljima omogućavaju učenje kroz zabavu, vrlo je važno implementirati ih u nastavu unatoč krutom planu i programu. Cilj je učenike naučiti tj. pripremiti ih za život, a odgovornost je na učiteljima da u tu svrhu stalno istražuju nova sredstva te time ulijevaju znanja novim generacijama.

## Popis literature

1. <http://www.kurikulum.hr/wp-content/uploads/2016/03/Informatika.pdf> , [citirano 28.07.2016.]
2. Prensky M.(2015.) “*Plan B*” *Education to Improve the World, The Emerging Alternative for K-12 Education; 2015*, [citirano 12.03.2016.] Dostupno na: <http://marcprensky.com/wp-content/uploads/2015/12/+Prensky-PlanB.pdf>
3. <https://scratch.mit.edu> [citirano 28.07.2016.]
4. <https://scratch.mit.edu/statistics/> [citirano 28.07.2016.]
5. [http://cty.jhu.edu/ctyonline/courses/computer\\_science/scratch\\_programming\\_elementary.html](http://cty.jhu.edu/ctyonline/courses/computer_science/scratch_programming_elementary.html) [citirano 28.07.2016.]
6. Bagge, P., (2015.), *How to Teach Primary Programming Using Scratch: Teacher's Handbook (Code-IT Primary Programming) A complete KS2 Computer Science study programme*, The University of Buckingham Press, UK
7. Melmoth, J., Dickins, R., Stowell, S., Nielsen, S., (2015.), *Coding for Beginners: Using Scratch*, Usborne Publishing Ltd, UK
8. Woodcock, J., Setford, S., *DK Workbooks: Coding in Scratch: Games Workbook*, DK Publishing (Dorling Kindersley); Csm Wkb edition, UK
9. <http://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf> [citirano 28.07.2016.]