

Digitalna kreativnost u učionici: izrada mini-igara u alatu „Microsoft MakeCode Arcade“ u 4. razredu osnovne škole

Primjer dobre prakse



Sažetak

Ovaj primjer dobre prakse prikazuje uporabu digitalnog alata *Microsoft MakeCode Arcade* u nastavi Informatike u 4. razredu osnovne škole. Cilj aktivnosti bio je razvoj računalnog razmišljanja, logičkog zaključivanja i digitalne kreativnosti izradom vlastite mini-igre. Nastava je organizirana kao projektni rad u paru ili manjoj skupini uz diferencirani pristup te primjenu formativnog i sumativnog vrednovanja. Učenici su pokazali visoku motivaciju, uspješnu primjenu programskih struktura te razvoj suradničkih i metakognitivnih vještina. Model rada je prenosiv i prilagodljiv različitim razinama složenosti te predstavlja primjer suvremene i inovativne nastave u kojoj vizualno programiranje omogućuje učinkovito razvijanje računalnog razmišljanja i digitalne kreativnosti.

1. Uvod

Razvoj računalnog razmišljanja i digitalnih kompetencija jedan je od temeljnih ciljeva suvremenog obrazovanja. Programiranje u nižim razredima osnovne škole pridonosi razvoju logike, analitičkog promišljanja i kreativnosti, ali istodobno može biti izazovno zbog apstraktnosti pojmova i različitih razina predznanja učenika. Kako bi se programiranje učenicima približilo na razumljiv i motivirajući način, u nastavi je primijenjen digitalni alat *Microsoft MakeCode Arcade*. Riječ je o vizualnom okruženju koje omogućuje izradu 2D igara korištenjem blok-programiranja, a takav pristup učenicima pruža trenutačnu povratnu informaciju, aktivno učenje i visoku razinu angažiranosti. Aktivnost je usklađena s ishodima kurikulumu nastavnog predmeta Informatika za razrednu nastavu te pridonosi razvoju digitalnih kompetencija i računalnog razmišljanja.

2. Ciljevi i odgojno-obrazovni ishodi

Aktivnost je provedena u dvama 4. razreda osnovne škole s ukupno 35 učenika (17 u jednom i 18 u drugom razredu) tijekom četiri školska sata.

Ostvareni ishodi:

B.4.1 – Učenik stvara program u vizualnom okruženju koristeći slijed, ponavljanje, odluku i ulazne vrijednosti.

(primjena: izrada miniigre u MakeCode Arcadeu korištenjem događaja, uvjeta, petlji i upravljanja likovima)

B.4.2 – Učenik rješava složenije logičke zadatke uporabom računala.

(primjena: rješavanje problema kroz izradu i testiranje igre, uočavanje pogrešaka (debugging) i prilagodbu programskog rješenja)

Cilj aktivnosti bio je osposobiti učenike za samostalno stvaranje funkcionalne mini-igre primjenom osnovnih programskih struktura te poticati razvoj suradničkih, kreativnih i refleksivnih vještina.

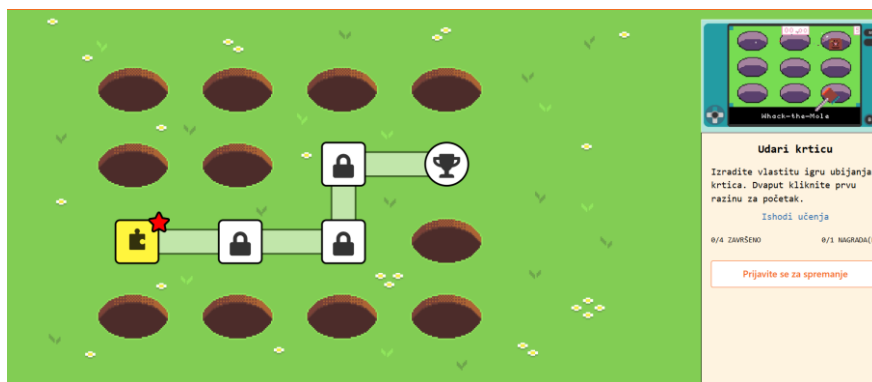
3. Opis provedbe aktivnosti

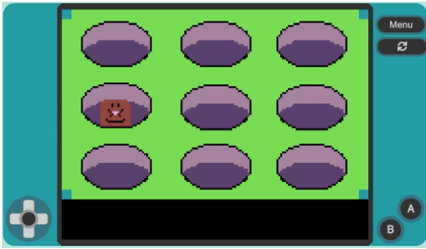

Aktivnost je bila organizirana kao projektni rad u kojem su učenici tijekom više nastavnih sati planirali, izrađivali, testirali i predstavljali vlastitu mini-igru. Prolazili su strukturiranim *Skillmap* vodičem unutar platforme *MakeCode Arcade*, a zatim su igru kreativno nadograđivali prema vlastitim idejama. Radili su u paru ili manjoj skupini uz dogovorene uloge, dok je svaki učenik program izrađivao na vlastitom uređaju što je omogućilo istodobni razvoj suradničkih i individualnih digitalnih vještina.

Tijekom rada učenici su:

- izrađivali *spriteove* (likove i pozadinu)
- koristili koordinate i slučajne pozicije
- primjenjivali uvjetne naredbe
- koristili petlje (*forever*) za ponavljanje radnji u igri
- dodavali događaje (*on button pressed*)
- uključivali brojač bodova i odbrojavanje vremena
- testirali i ispravljali pogreške (*debugging*)
- dodavali kreativne elemente (zvuk, animacije, nove razine)



Nastava je bila organizirana kao kombinacija demonstracije, vođenog otkrivanja i problemskog učenja. Učitelj je imao ulogu mentora koji potiče promišljanje, donošenje odluka i samostalno rješavanje problema te učenicima pruža podršku poticajnim pitanjima i usmjeravanjem tijekom procesa programiranja.



Aktivnost	Lov na krtice (18 min)
	<p>Naučite koristiti MakeCode Arcade i dodajte sprite pokretnog lika u svoj projekt.</p>
<p>Korišteni blokovi</p>	

```

on start
  set background image to [grid]
  set mySprite to sprite of kind Enemy
  on game update every 1000 ms
    move sprite mySprite in grid
    
```


Aktivnost	Vrijeme čekića (12 min)
	<p>Dodajte gumeni čekić kako biste označili krticu i zaradili bodove!</p>
<p>Korišteni blokovi</p>	

```

on start
  set background image to [grid]
  set mySprite to sprite [mole] of kind Enemy
  set myHammer to sprite [hammer] of kind Player
  move myHammer on screen with speed fast
  start countdown 15 (s) and game over high score effect confetti

on game update every 1000 ms
  move sprite mySprite in grid

on sprite of kind Player overlaps otherSprite of kind Enemy
  change score by 1
  move sprite mySprite in grid
    
```



Aktivnost	Animiraj se (15 min)
	<p>Dodajte zvuk i animaciju kako biste usavršili svoju igru!</p>
<p>Korišteni blokovi</p>	<pre> play sound knock animate myHammer frames [hammer] interval (ms) 50 loop OFF add label "Whack-the-Mole" to bottom of window </pre>

```

on start
  set background image to [grid]
  set mySprite to sprite [mole] of kind Enemy
  set myHammer to sprite [hammer] of kind Player
  move myHammer on screen with speed fast
  start countdown 15 (s) and game over high score effect confetti
  add label "Whack-em-Mole" to bottom

on game update every 1000 ms
  move sprite mySprite in grid

on sprite of kind Player overlaps otherSprite of kind Enemy
  change score by 1
  move sprite mySprite in grid
  play sound ba ding
  animate myHammer
    frames [hammer]
    interval (ms) 200
    loop OFF
    
```

Aktivnost	Animiraj se (15 min)
	<p>Nekoliko jednostavnih promjena će vas u tren oka natjerati da igrate potjeru!</p>
<p>Korišteni blokovi</p>	

```

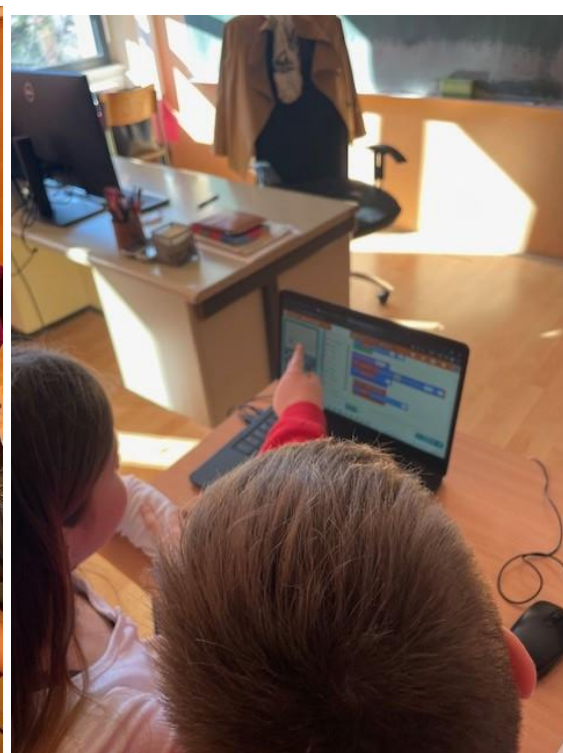
on start
  show long text "Player 1! Use the arrow keys to move the hammer..."
  set background image to [grid]
  set myMole to sprite [mole] of kind Enemy
  set myHammer to sprite [hammer] of kind Player
  move myHammer on screen with speed fast
  start countdown 15 (s) and game over multiplayer
  add label "Whack-the-Mole" to bottom of window

on game update every 1000 ms
  player 2 score 1 for no grid hop

on sprite of kind Player overlaps othersprite of kind Enemy
  change score by 1
  move sprite myMole in grid
  play sound knock
  animate myHammer
  frames 1
  interval (ms) 50
  loop

on player 2 A button pressed
  move sprite myMole in grid
  
```

Primjeri učničkih aktivnosti





4. Trajanje i organizacija rada

Aktivnost je planirana i provedena tijekom četiriju nastavnih sati što se pokazalo optimalnim za učenike 4. razreda koji su učili programirati u blokovima.

1. sat — Uvod i izrada osnovne strukture igre (45 min)

Učenici su se upoznali s okruženjem *MakeCode Arcade*, izradili prve *spriteove* i uveli osnovne događaje (*on start, on overlap*). Pratili su strukturirani vodič i oblikovali temeljne elemente igre *Whack the Mole*.

2. sat — Logički elementi i ponašanje likova (45 min)

Dodavali su uvjetne naredbe, slučajne pozicije, brojač bodova i odbrojanje vremena. Testirali su funkcionalnost, uspoređivali različite pristupe i ispravljali pogreške u logici igre.

3. sat — Proširenja i kreativni elementi (45 min)

Uvodili su dodatne razine težine, zvukove, animacije i prilagođene efekte. Napredniji učenici istraživali su *JavaScript* prikaz koda i optimizirali strukturu igre.

4. sat — *Debugging*, prezentacija i vrednovanje (45 min)

Učenici su sustavno testirali funkcionalnost, dorađivali logiku i rješavali pogreške. Nakon toga, predstavili su svoje mini-igre, obrazložili izbor programskih strategija i sudjelovali u vrednovanju prema unaprijed pripremljenoj rubrici.

Nakon prezentacija, ispunili su obrazac za samorefleksiju kao dio vrednovanja za učenje.

Ovakva organizacija rada omogućila je postupno usvajanje programerskih koncepata, jasnu gradaciju složenosti te dovoljno vremena za eksperimentiranje, kreativnost i razvoj računalnog razmišljanja.

Napomena o trajanju

Planiranje aktivnosti tijekom četiriju nastavnih sati omogućilo je postupnu gradaciju složenosti, dovoljno vremena za izradu osnovne strukture igre te za testiranje i ispravljanje pogrešaka (*debugging*). Posebna je pažnja posvećena učenicima kojima je bila potrebna dodatna podrška jer im je trebalo više vremena za objašnjavanje blokova, razumijevanje logičkih odnosa u igri i vođeno usmjeravanje prilikom rješavanja zadataka. Napredniji su učenici imali priliku samostalno nadograđivati igru dodatnim razinama, animacijama, efektima i složenijim uvjetima pri čemu su se nadogradnje razlikovale među skupinama ovisno o njihovim idejama, interesima i mogućnostima. Dodatno vrijeme bilo je potrebno i za prezentaciju izrađenih mini-igara te učeničko objašnjavanje programskih naredbi i odabranih strategija čime su razvijali komunikacijske vještine i sposobnost argumentiranja vlastitih rješenja. Takva je organizacija rada omogućila svim učenicima da u skladu sa svojim mogućnostima razvijaju programersko razmišljanje, kreativnost i vještine rješavanja problema, a ne samo izradu završnog proizvoda.

5. Povezanost aktivnosti i odgojno-obrazovnih ishoda

Aktivnost je izravno povezana s odgojno-obrazovnim ishodima kurikuluma Informatike za 4. razred osnovne škole. Učenici su izradom vlastite mini-igre primijenili osnovne programske strukture poput slijeda, ponavljanja i uvjeta čime su ostvarili ishode B.4.1 i B.4.2. Istodobno su razvijali računalno razmišljanje, sposobnost rješavanja problema i suradničke vještine.

Tablica 1. Povezanost aktivnosti i odgojno-obrazovnih ishoda

Aktivnost učenika	Razvijene vještine	Odgojno-obrazovni ishod
Izrada osnovnih elemenata igre (<i>spriteovi</i> , pozadina, događaji <i>on start</i> , <i>on overlap</i>).	razumijevanje slijeda, inicijalnih stanja i osnovnih programskih struktura	B.4.1 – stvara program u vizualnom okruženju koristeći slijed, ponavljanje, odluku i ulazne vrijednosti
Uvođenje uvjetnih naredbi, slučajnih pozicija, brojača bodova i odbrojanja vremena.	rješavanje programskih problema, logičko zaključivanje, algoritamsko razmišljanje	B.4.2 – rješava složenije logičke zadatke uporabom računala
Testiranje i ispravljanje pogrešaka (<i>debugging</i>).	analiza pogrešaka, evaluacija rješenja, prilagodba strategije	B.4.2
Proširenje igre dodavanjem zvuka, animacija, novih razina težine.	kreativnost, projektno učenje, proširivanje koda	B.4.1
Suradnja u paru ili skupini tijekom izrade igre..	komunikacija, dijeljenje ideja, argumentiranje rješenja	B.4.1 i B.4.2 (ovisno o fazi rada i zadatku)
Prezentacija mini-igre i objašnjavanje vlastitog koda	metakognitivne vještine, refleksija o procesu učenja	B.4.2

6. Međupredmetne teme

Aktivnost doprinosi ostvarivanju pojedinih međupredmetnih tema:

Uporaba informacijske i komunikacijske tehnologije – učenici koriste digitalni alat za izradu vlastite igre, istražuju mogućnosti vizualnog programiranja i razvijaju digitalne vještine.

Učiti kako učiti – testiranjem i ispravljanjem pogrešaka učenici razvijaju strategije rješavanja problema, reflektiraju o vlastitom procesu učenja i prilagođavaju pristupe programiranju.

Poduzetništvo – učenici osmišljavaju ideje za vlastitu igru, kreativno nadograđuju rješenja i preuzimaju odgovornost za realizaciju projekta.

7. Diferencijacija i individualizacija

U nastavi je bio primijenjen diferencirani pristup kako bi se zadovoljile različite razine sposobnosti, predznanja i tempa rada učenika.

Učenici kojima je bila potrebna dodatna podrška:

- radili prema detaljnim, jasno strukturiranim uputama
- dobivali dodatna pojašnjenja blokova
- fokusirali se na osnovnu funkcionalnost igre

Učenici koji brže napreduju:

- dodavali dodatne razine igre
- uvodili složenije uvjete (npr. negativne bodove, dodatne likove)
- istraživali *JavaScript* prikaz koda
- preuzimali ulogu „digitalnih mentora“

Takav pristup omogućio je svakom učeniku napredak u skladu sa svojim mogućnostima te poticao razvoj samostalnosti, odgovornosti i suradničkih vještina.

8. Vrednovanje

Vrednovanje je bilo planirano kao kontinuirani proces koji je pratio učenike tijekom cijelog projekta.

Vrednovanje za učenje

Tijekom aktivnosti učitelj je pružao redovite povratne informacije, postavljao poticajna pitanja i usmjeravao učenike na prepoznavanje logičkih odnosa u igri te na samostalno otkrivanje rješenja. Simulator u alatu omogućio je trenutačnu provjeru funkcionalnosti što je učenicima pomoglo uočavanje posljedica svojih odabira i prema potrebi prilagodbu koda.

Vrednovanje kao učenje

Učenici su razvijali metakognitivne vještine raspravom, međusobnim objašnjavanjem postupaka i usporedbom različitih strategija programiranja. Poseban naglasak bio je na osvještavanju procesa rješavanja problema, odnosno učenici su opisivali svoj postupak, objašnjavali u paru ili skupini kako su došli do rješenja i prepoznavali elemente koje bi mogli poboljšati.

Nakon prezentacije radova, ispunili su obrazac za samorefleksiju (*checklistu*) u kojem su procjenjivali razumijevanje logike igre, uočavanje pogrešaka, promjenu strategije, razinu kreativnosti i kvalitetu suradnje.

Primjeri učeničkih refleksija:

U svojim su osvrtima učenici isticali da im je rad u *MakeCode Arcade* pomogao jasnije razumjeti što se događa „u pozadini“ igre koju sami stvaraju:

„Shvatio sam da mi blokovi pokazuju točno što se događa kad kliknem na tipku.“

Naglašavali su važnost postupnog testiranja i uočavanja pogrešaka:

„Najviše mi je pomoglo kad smo svaku promjenu odmah testirali, tako sam brže vidio gdje je greška.“

Prepoznali su i važnost promjene strategije u rješavanju zadataka:

„Morali smo promijeniti način kako smo napravili bodove jer stari način nije dobro radio.“

Uočili su i vrijednost suradnje:

„Kad sam objašnjavao drugom učeniku što sam napravio, i meni je postalo jasnije.“

Na kraju su naglasili da im je *debugging* postao koristan, a ne frustrirajući proces:

„Na početku sam mislio da su greške loše, ali zapravo su mi pokazale što trebam popraviti.“

Prethodne refleksije ukazuju na razvoj računalnog razmišljanja, strategijskog programiranja i samoreguliranog učenja kod učenika što predstavlja jedan od ključnih ciljeva programiranja u razrednoj nastavi informatike.

OBRAZAC ZA SAMOREFLEKSIJU UČENIKA (*MakeCode Arcade* – 4. razred)

Ime i prezime: _____ Datum: _____

|

Razumijevanje

- Razumijem što se događa "u pozadini" igre koju sam napravio/la.
- Razumijem što radi većina blokova koje sam koristio/la.

Testiranje i pogreške

- Nakon svake promjene testirao/la sam igru.
- Primijetio/la sam barem jednu pogrešku (**bug**).
- Znam objasniti kako sam tu pogrešku pronašao/la.
- Znam objasniti kako sam je ispravio/la.

Promjena strategije

- Morao/la sam promijeniti početni način rješavanja problema.
- Promjena strategije pomogla mi je da igra bolje radi.

Kreativnost

- Dodao/la sam barem jedan kreativni element (zvuk, animaciju, pozadinu...).
- Zadovoljan/na sam kako izgleda moja igra.

Suradnja

- Suradnja u paru/skupini pomogla mi je u rješavanju zadatka.
- Kada sam drugome objašnjavao/la što sam napravio/la, i meni je postalo jasnije.

Danas sam najviše naučio/naučila da...

Vrednovanje naučenog

Završna mini-igra bila je vrednovana prema unaprijed pripremljenoj rubrici koja je obuhvaćala primjenu programskih struktura, funkcionalnost igre, rješavanje problema i suradnju. Učenici su svoje igre predstavili pred razredom i obrazložili odabrana programska rješenja čime su dodatno razvijali komunikacijske i analitičke vještine. Rubrika je omogućila objektivan uvid u razinu ostvarenosti ishoda B.4.1 i B.4.2 te jasnu analizu završnog rada.

Rubrika je obuhvaćala sljedeće kriterije:

- primjenu programskih struktura (uvjeti, petlje, događaji)
- funkcionalnost i logiku igre
- kreativnost i dodatne elemente
- sposobnost rješavanja problema
- suradnju u paru ili skupini

Tablica 2. Rubrika za vrednovanje učeničkih projekata

Kriterij	5 – Izvrsno	4 – Vrlo dobro	3 – Dobro	2 – Djelomično ostvareno	1 – Početna razina
Primjena programskih struktura	Primjenjuje više struktura (uvjeti, događaji, brojači, slučajnost, petlje). Kod je logičan, povezan i kreativno nadograđen.	Primjenjuje više osnovnih struktura bez pogrešaka; kod je jasan i funkcionalan.	Primjenjuje osnovne strukture; igra uglavnom radi uz manje pogreške.	Koristi samo jednostavne strukture; dio logike nije povezan ili ne radi ispravno.	Ne primjenjuje strukture samostalno; igra nije funkcionalna bez pomoći.
Funkcionalnost igre	Igra radi stabilno, pravilno reagira, ima jasna pravila i završetak; svi elementi funkcioniraju.	Igra se može normalno odigrati, a manje pogreške ne utječu na pravila ni tijek igre.	Igra se može odigrati, ali manji nedostaci utječu na kvalitetu igranja.	Igra djelomično radi; postoje veće logičke pogreške.	Igra se ne može igrati; nedostaju ključni elementi.
Rješavanje problema (debugging)	Samostalno testira, pronalazi i ispravlja pogreške; mijenja strategiju i objašnjava postupke.	Uglavnom samostalno ispravlja pogreške; treba minimalnu podršku.	Prepoznaje i ispravlja pogreške uz podršku.	Uočava pogreške tek uz poticaj; ispravci su ograničeni.	Ne prepoznaje niti ispravlja pogreške bez potpune pomoći.
Suradnja	Aktivno surađuje, preuzima inicijativu, potiče druge i uvažava prijedloge.	Dobro surađuje, dijeli zadatke i komunicira učinkovito.	Suradnja je stabilna, ali povremeno nedovoljno aktivna.	Suradnja je povremena; učenik rijetko doprinosi timu.	Ne sudjeluje aktivno u radu tima ili se u potpunosti oslanja na druge učenike.

9. Rezultati i učenička postignuća

Tijekom provedbe aktivnosti uočena je visoka razina motivacije učenika. Većina je uspješno primijenila uvjetne naredbe i događaje te samostalno ispravljala pogreške u kodu.

Učenici su pokazali:

- razvoj računalnog razmišljanja
- veću samostalnost u radu
- poboljšanu suradnju u paru ili skupini
- sposobnost obrazlaganja vlastitih rješenja

Obrasci za samorefleksiju pokazuju da su učenici prepoznali napredak u razumijevanju uvjeta i petlji te izrazili želju za izradom novih igara. Nekoliko naprednijih učenika proširilo je zadatak izradom još jedne mini-igre.

10. Refleksija učitelja

Primjena *MakeCode Arcade* alata pokazala se iznimno učinkovitom u povećanju motivacije učenika i razumijevanja programskih koncepata. Vizualno okruženje i trenutačna povratna informacija omogućili su učenicima eksperimentiranje i učenje na vlastitim pogreškama.

Posebno se učinkovitim pokazala kombinacija projektne nastave, suradničkog rada, diferenciranog pristupa i strukturiranog formativnog vrednovanja. Vizualno programiranje u blokovima smanjuje strah od pogreške i omogućuje postupno usvajanje programskih struktura poput petlji, uvjeta i događaja.

U budućnosti bi se aktivnost mogla proširiti izradom tematskih igara povezanih s drugim nastavnim predmetima.

11. Zaključak i prenosivost

Primjena *Microsoft MakeCode Arcade* alata u razrednoj nastavi predstavlja suvremenu, inovativnu i pedagoški utemeljenu praksu. Aktivnost omogućuje razvoj digitalnih kompetencija, kreativnosti, logičkog zaključivanja i suradničkih vještina.

Opisani primjer lako se prilagođava različitim razredima i razinama složenosti te može služiti kao model dobre prakse u razvoju računalnog razmišljanja i digitalne kreativnosti učenika. Vizualno programiranje učenicima omogućuje intuitivno razumijevanje osnovnih programskih koncepata i potiče njihovu motivaciju za daljnje učenje programiranja.

Aktivnost se može povezati i s drugim nastavnim predmetima, primjerice matematikom (logičko zaključivanje, koordinatni sustav) ili hrvatskim jezikom (osmišljavanje priče i scenarija igre) čime se dodatno potiče međupredmetno učenje. Ovakav pristup potiče aktivno sudjelovanje učenika i razumijevanje programskih koncepata kroz iskustveno učenje, stvaranje i rješavanje problema.

Digitalni alat korišten u nastavi

Microsoft MakeCode Arcade - <https://arcade.makecode.com/>

Izvori

Ministarstvo znanosti i obrazovanja. (2018). *Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*.

Microsoft. (2024). *MakeCode Arcade – platforma za učenje programiranja kroz izradu igara*. Dostupno na: <https://arcade.makecode.com/>

Ključne riječi: računalno razmišljanje, *MakeCode Arcade*, vizualno programiranje, digitalne kompetencije, projektna nastava, razredna nastava informatike, digitalna kreativnost