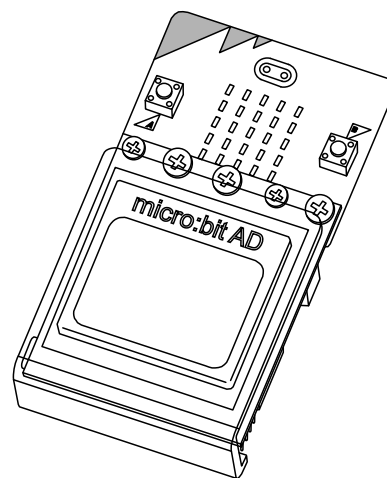


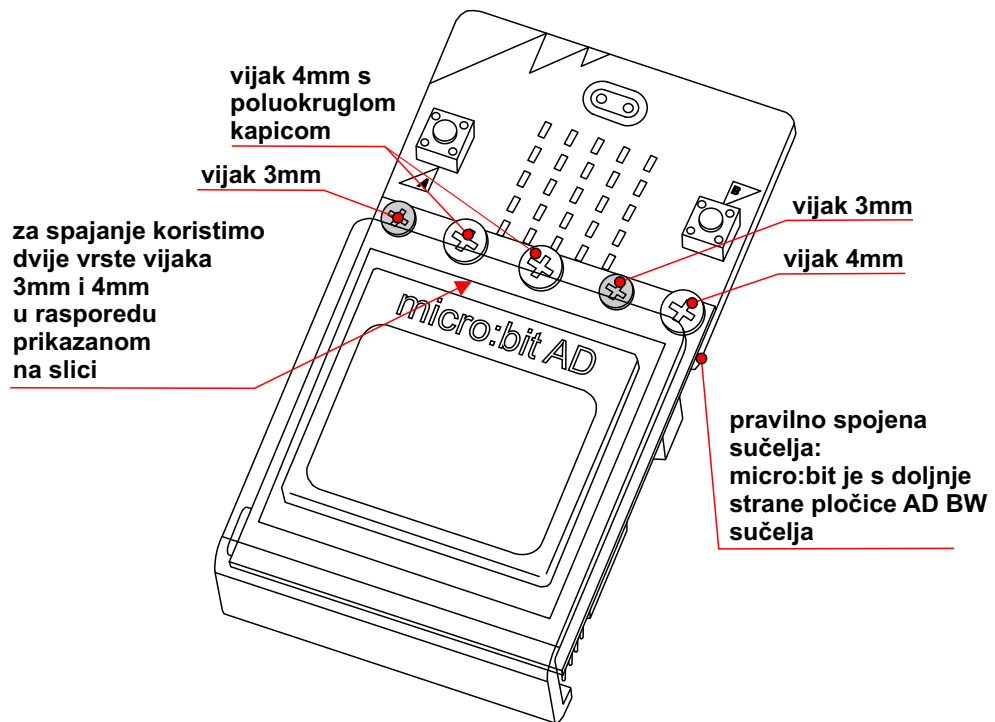
MakeCode programiranje

micro:bit AD BW



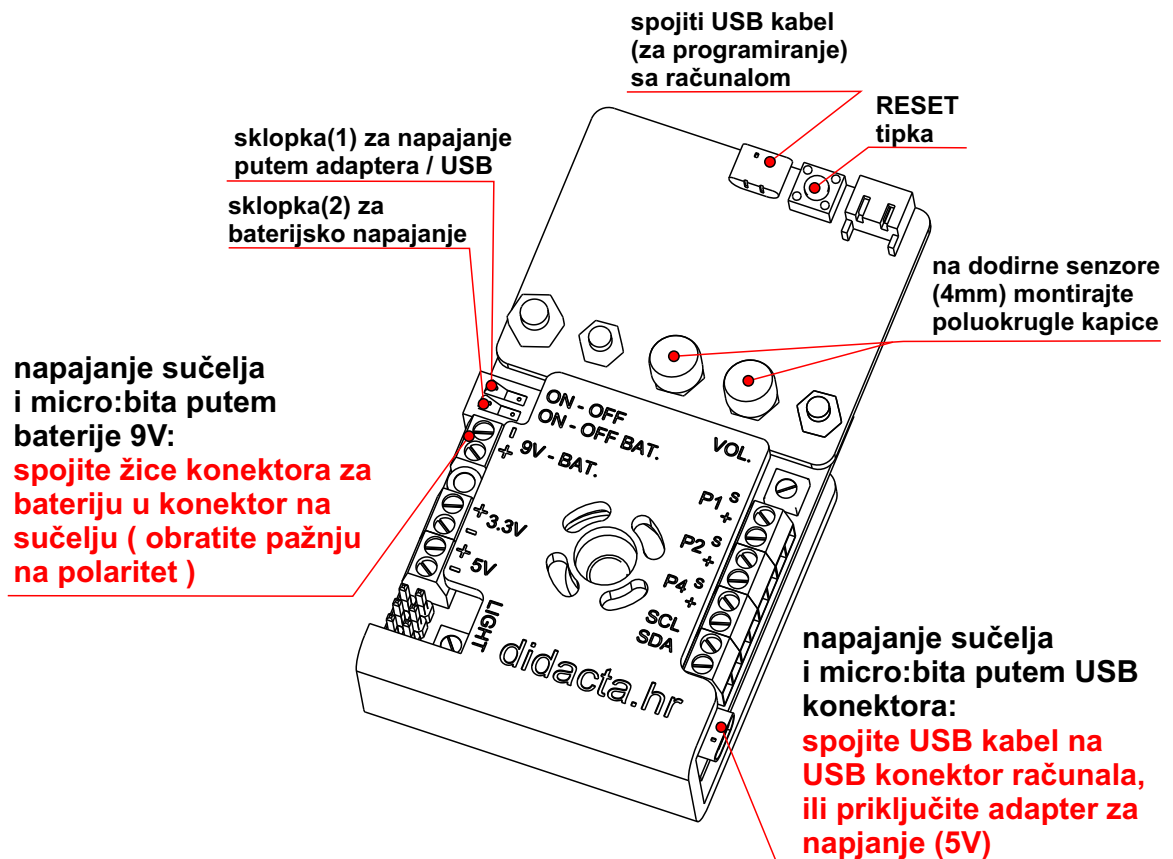
1. POČETAK

1.1. Spajanje micro:bita i AD sučelja vijcima



Manje vijke koristimo na pozicijama na kojima bi moglo doći do oštećenja elemenata na micro:bit pločici, a koji se nalaze blizu otvora za spajanje.

1.2. napajanje micro:bita i AD sučelja putem baterije, adaptera ili USB porta (PC)

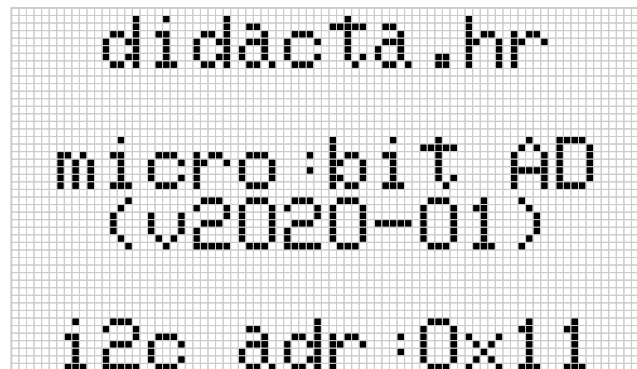


1.3. Pokretanje sučelja micro:bit AD

Nakon pokretanja, na ekranu sučelja micro:bit AD BW će se ispisati tekst prikazan na Slici 1.

Sučelje je spremno za rad.

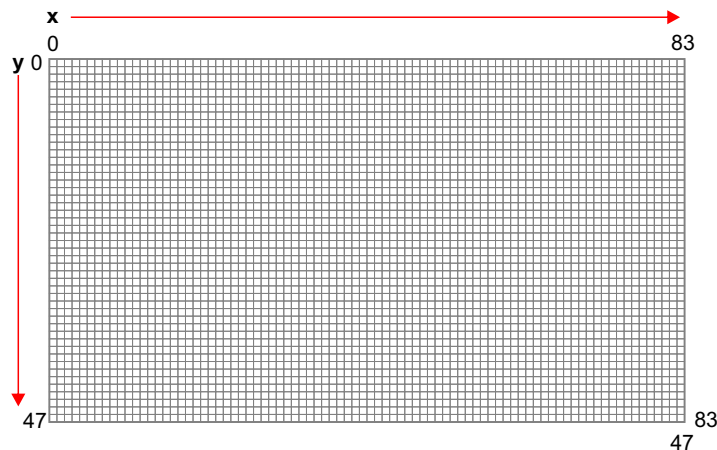
Ako je već učitana program u micro:bit, za pokretanje trebate pritisnuti tipku RESET na micro:bitu.



slika 1.

1.4. Ekran sučelja

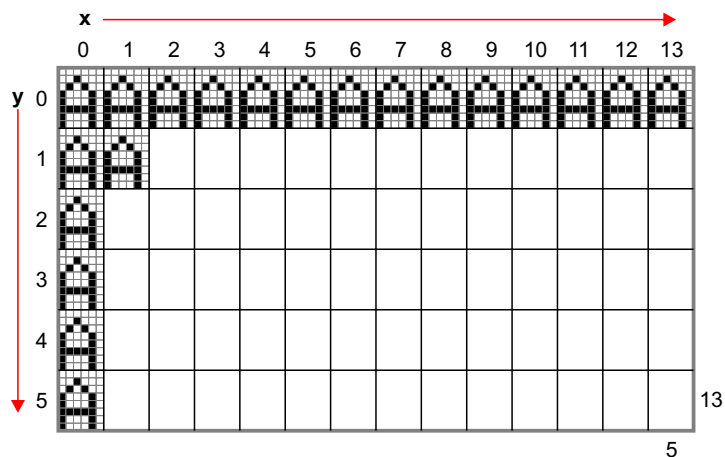
Micro:bit AD BW sučelje ima crno-bijeli ekran grafičke rezolucije 48 x 84 piksela (Slika 2.).



Slika 2. Grafički mod - rezolucija 48 x 84.

Za ispis teksta (tekst mod) se koristi rezolucija koja odgovara veličini fonta (znaka). Veličina standardnog tekstualnog znaka (slova - fonta je 7 x 5 piksela) je 8 x 6 piksela sa pikselima za razmak. Zato je rezolucija za ispis teksta 6 x 14 znakova (Slika 3.). Prilikom izrade programa moramo voditi računa koje programske naredbe koristimo i za koji mod rada su one namjenjene.

Grafičke funkcije koriste grafičku rezoluciju (linija, kružnica, ravokutnik,...) , a tekstualni mod se koristi kod standardnog ispisa teksta (ne grafičkog) i kod definiranja ekrana za igru, te pozicioniranja objekta igrača.



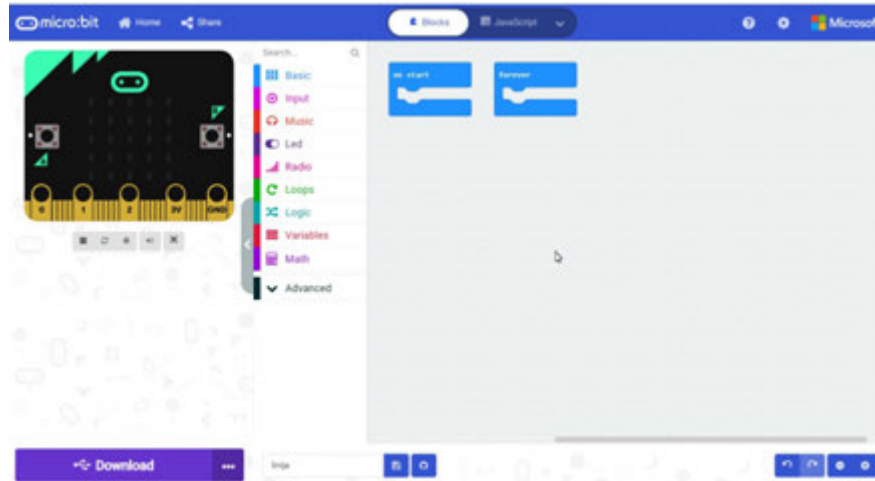
Slika 3. Tekstualni mod - rezolucija 6 x 14.

2. SUČELJE MAKECODE I BIBLIOTEKA

2.1. Pokretanje sučelja za programiranje i učitavanje biblioteke

Pokrenite MakeCode sučelje za programiranje u vašem internet eksploreru (link):

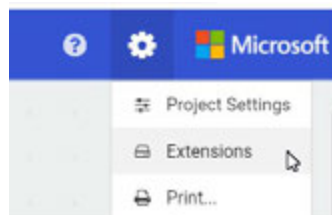
<https://makecode.microbit.org/#editor>




Slika 4. Makecode početni ekran sučelja

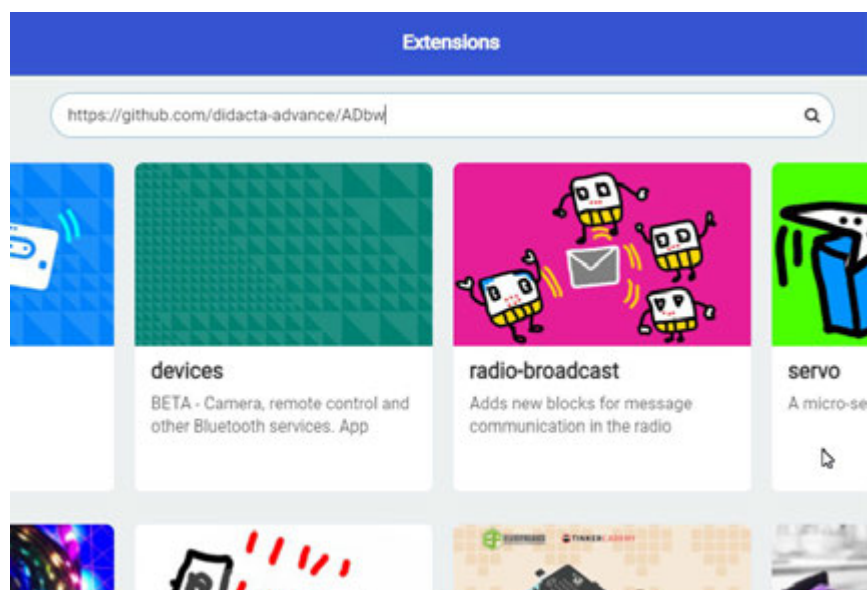
Učitajte biblioteku za micro:bit AD sučelje:

<https://github.com/didacta-advance/ADbw>



Slika 5. Odabir Extensions - za upis adrese biblioteke

Kliknite na **setup** (), a zatim odaberite **Extensions**.



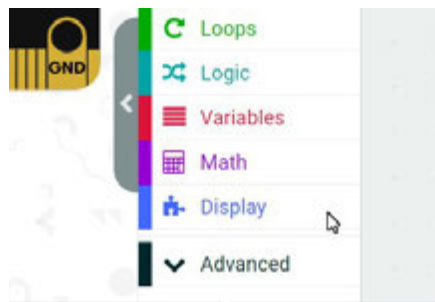
Slika 6. Upis adrese biblioteke (ekran može izgledati i drugačije)

Upišite adresu biblioteke.



Slika 7. Odabrana biblioteka

Kliknite na prozor biblioteke **displaylib** za početak učitavanja (Slika 7.).



Slika 8. Biblioteka u izborniku makecode sučelja

Nakon učitavanja biblioteke, naziv biblioteke **Display** se treba pojaviti u izborniku (Slika 8.).

3. PROGRAMIRANJE - OSNOVNE FUNKCIJE

3.1. Prvi program - ispis teksta "HELLO" - funkcija TEXT

U prvom programu koristimo funkciju **TEXT** (slika 9.).



Slika 9. Funkcija RESET PROGRAM



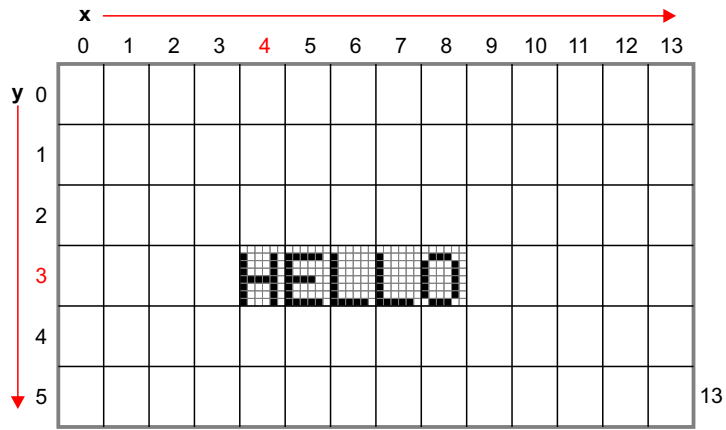
Slika 10. Funkcija TEXT (tekst mod)

Na početku svakog programa dobro je koristiti funkciju **RESET PROGRAM** koja briše vrijednosti koje je program sučelja micro:bit AD koristio u prethodnom radu programa na micro:bitu.

Funkciju **TEXT** umetnemo u već postojeći programski blok **on start** koji se nalazi na radnoj podlozi sučelja. U polje "" upišite tekst HELLO i poziciju ispisa (x,y) na ekranu (tekst mod) prema primjeru na Slici 11. Učitajte program u micro:bit putem komande **DOWNLOAD**. Na ekranu bi se trebao ispisati tekst **HELLO** na poziciji upisanih vrijednosti (Slika 12.).



Slika 11. Program za ispis teksta HELLO



Slika 12. Ispis na ekranu

Ispis na ekranu nakon pokretanja prvog programa Slika 12.

Upisom vrijednosti 2 ili 3 u polje **size** možete ispisati tekst veći od standardne dimenzije (Slika 13.).

```

on start
  RESET PROGRAM
  TEXT: "HELLO" - on col x(0-13) 2 , row y(0-5) 1 size (0-3) 2 color
  
```

Slika 13. Ispis na ekranu

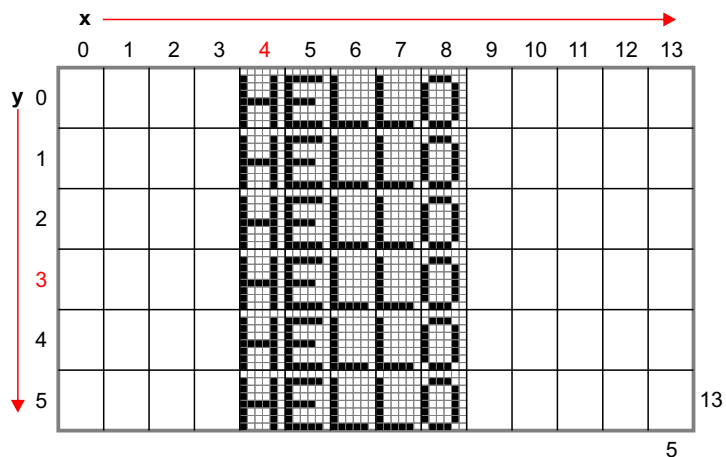
3.2. Višestruki ispis teksta putem funkcije repeat

U ovom programu koristimo funkciju za ponavljanje - **repeat** i varijablu **brojac** koju koristimo kao y vrijednost u funkciji **TEXT**.

```

on start
  RESET PROGRAM
  set brojac to 0
  repeat 6 times
    do
      TEXT: "HELLO" - on col x(0-13) 4 , row y(0-5) brojac size (0-3) 0 color
      change brojac by 1
  
```

Slika 14. Višestruki ispis teksta



Slika 15. Ispis na ekranu programa Slika 14.

3.3. Ispis teksta na dvije pozicije i brisanje ekrana

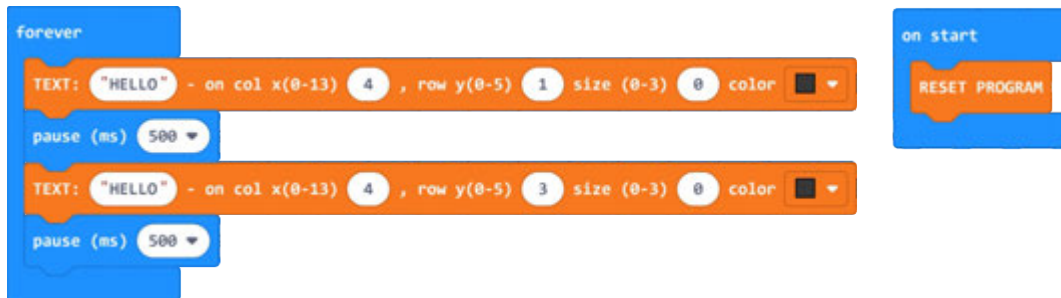
U ovom programu koristimo funkciju **TEXT** i funkciju za brisanje ekrana **CLEAR SCREEN** (Slika 15.).



Slika 16. Funkcija CLEAR SCREEN - brisanje ekrana

U bloku **on start** ostavimo funkciju **RESET PROGRAM**.

Funkciju **TEXT** premjestimo u blok **forever** (zauvijek) i upišemo vrijednosti prema primjeru. Nakon prve funkcije dodamo **pauzu** od pola sekunde (500 ms). To ponovimo još jedan puta i upišemo druge vrijednosti u novu funkciju, prema primjeru na slici 15.



Slika 17. Program za ispis teksta HELLO na dvije pozicije

Dodajte u program funkciju za brisanje ekrana - **CLEAR SCREEN** i učitajte program.



Slika 18. Program za ispis teksta HELLO na dvije pozicije

U čemu je razlika kod prikaza na ekranu ?

Maknite **CLEAR SCREEN** funkciju iz programa. Upišite jednake vrijednosti za poziciju u obe funkcije **TEXT**, te drugoj funkciji promijenite boju ispisa.

Koji je rezultat ove promijene ?

3.4. Ispis teksta u grafičkom modu

U ovom programu koristimo funkciju **TEXT** za ispis teksta u grafičkoj rezoluciji (grafički mod Slika 2.). Putem ove funkcije možemo ispisati tekst na bilo kojoj poziciji na ekranu.

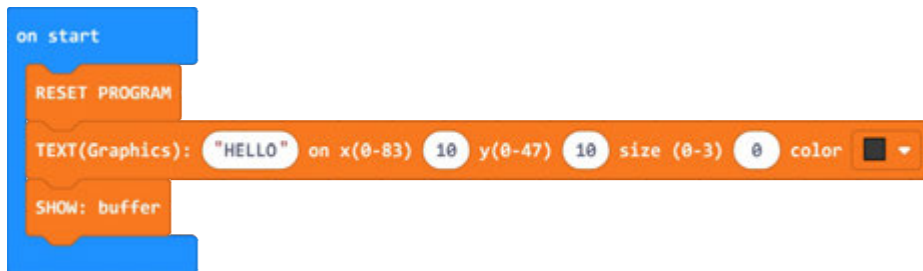
Ova funkcija ne ispisuje tekst direktno na ekran već u pomoćnu memoriju (**BUFFER**), zato se nakon jedne ili više **TEXT(Graphics)** funkcija mora izvršiti funkcija **SHOW: buffer** koja prikazuje zapis iz pomoćne memorije na ekranu.



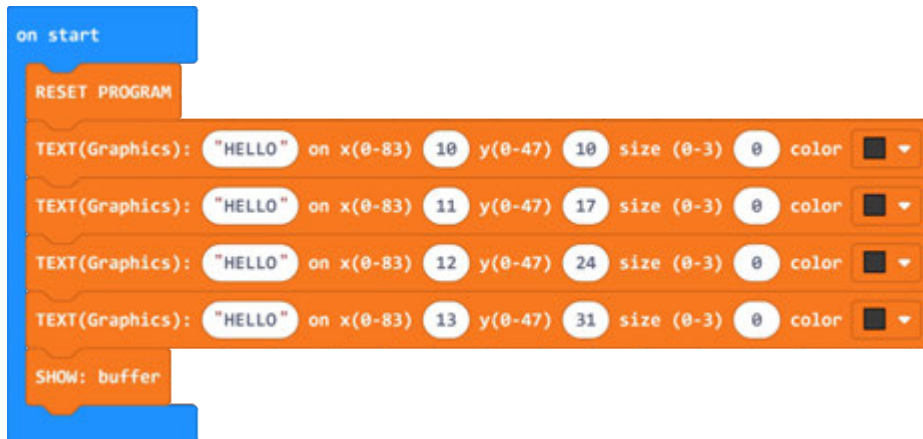
Slika 19. Funkcija CLEAR SCREEN - brisanje ekrana



Slika 20. Funkcija za ispis pomoćne memorije na ekranu

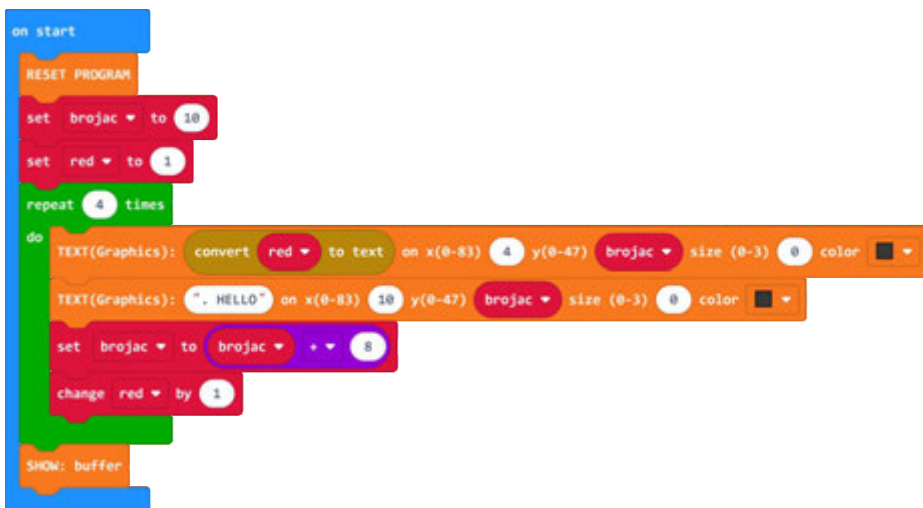


Slika 21. Primjer program za ispis jedne linije teksta u grafičkom modu



Slika 22. Primjer program za ispis više linija teksta u grafičkom modu

Primjer programa koji ispisuje numeričke vrijednosti na određenoj poziciji putem TEXT funkcije.



Slika 23. Primjer program za ispis više linija teksta u grafičkom modu

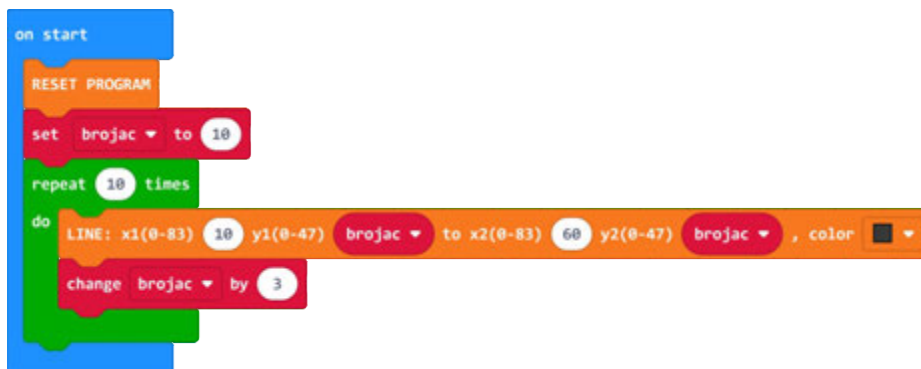
3.5. Iscrtavanje linije

U ovom programu koristimo funkciju **LINE** za iscrtavanje linije na ekranu u grafičkom modu. Kod linije trebamo odrediti početnu (x1,y1) i završnu (x2,y2) točku na ekranu. Liniju možemo iscrtati u crnoj ili bijeloj boji. Ako već iscrtanu liniju crne boje želimo izbrisati trebamo iscrtati na istu poziciju liniju u bijeloj boji.

Isprobajte!



Slika 24. Iscrtavanje linije u crnoj boji



Slika 25. Iscrtavanje više linija u crnoj boji



Slika 26. Iscrtavanje više linija u crnoj boji uz uporabu funkcije **PICK RANDOM**

3.6. Iscrtavanje kružnice ili kruga

Kružnica ili krug nisu pravilnog oblika radi rezolucije ekrana.

U ovim primjerima koristimo funkciju **CIRCLE** za iscrtavanje kružnice, u grafičkom modu. Kod kružnice trebamo odrediti poziciju središta (x,y) kružnice na ekranu i radijus. Kružnicu možemo iscrtati u crnoj ili bijeloj boji. Ako već iscrtanu kružnicu crne boje želimo izbrisati, trebamo iscrtati na istu poziciju kružnicu bijelom boji. Za iscrtavanje kruga koristimo se bojom ispunje (**filled with**).

Isprobajte!



Slika 27. Funkcija za iscrtavanje kružnice ili kruga



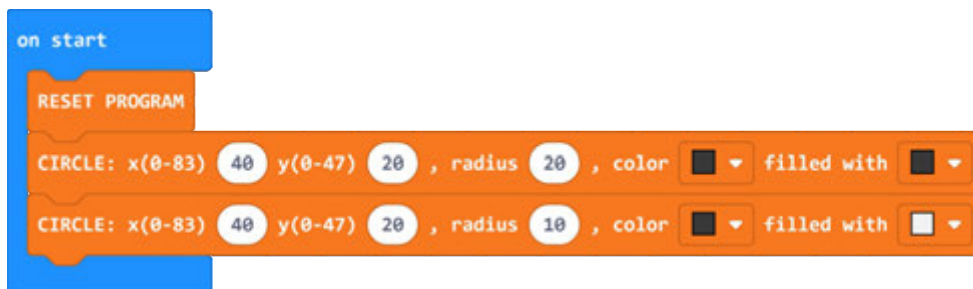
Slika 28.

Iscrtavanje kružnice radijusa 20 piksela na poziciji x = 40, y = 20 Slika 28.



Slika 29.

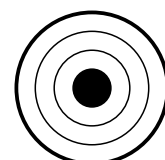
Iscrtavanje nekoliko kružnica različitog radijusa na istoj na poziciji x = 40, y = 20 (Slika 29.).



Slika 30.

Iscrtavanje bijelog kruga radijusa 10 unutar crnog kruga radijusa 20 piksela na poziciji x = 40, y = 20 (Slika 30.).

Za vježbu nacrtajte metu sa vanjskim debelim rubom i centrom u crnoj boji.



3.7. Iscrtavanje pravokutnika

U ovim primjerima koristimo funkciju **RECTANGLE** za iscrtavanje pravokutnika u grafičkom modu. Kod pravokutnika trebamo odrediti poziciju lijevog gornjeg kuta (x,y), širinu (0-83) i visinu (0-47). Pravokutnik možemo iscrtati u crnoj ili bijeloj boji. Ako već iscrtani pravokutnik crne boje želimo izbrisati, trebamo iscrtati, na istu poziciju, pravokutnik bijele boje. Pravokutnik može biti iscrtan samo linijama ili popunjen bojom (filled with).

Isprobajte!



Slika 31. Funkcija za iscrtavanje pravokutnika



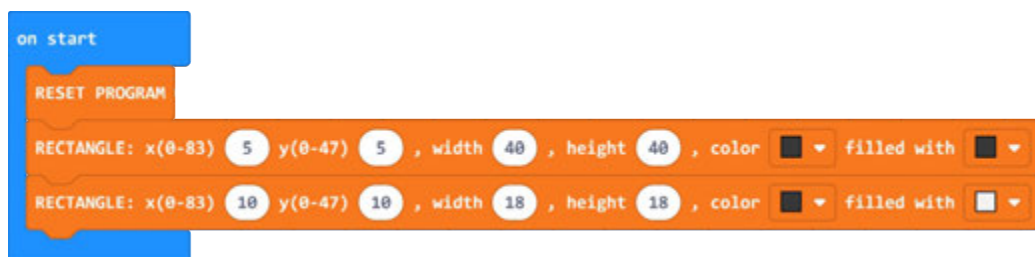
Slika 32.

Iscrtavanje pravokutnika na poziciji x = 5, y = 5 širine 73 i visine 37 piksela (Slika 32.).



Slika 33.

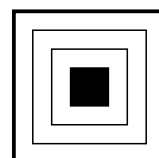
Iscrtavanje nekoliko pravokutnika različite pozicije i veličine (Slika 33.).



Slika 34.

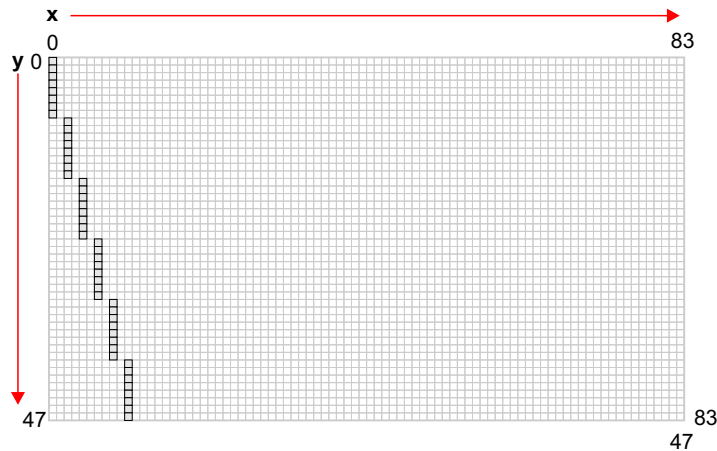
Iscrtavanje bijelog pravokutnika širine i visine 18, unutar crnog pravokutnika širina i visine 40 piksela (Slika 34.).

Za vježbu nacrtajte pravokutnumetu sa vanjskim debelim rubom i centrom u crnoj boji.



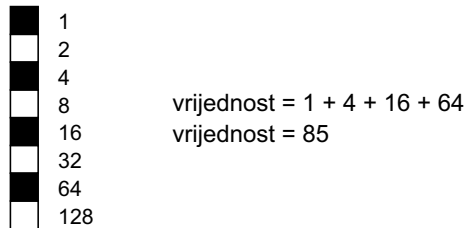
3.8. Bojanje ekrana - PAINT display

Funkcija PAINT ispunjava (boji) ekran bajtovima vrijednosti upisane u polje **color** (boja). Ekran se ispunjava bajtovim koji su položeni okomito kao na Slici 35.



Slika 35. Ispis bajtova memorije ekrana

Primjer ispune (bojanja) ekrana linijama u razmaku od jednog piksela. Izračun vrijednosti boje (bajta) možete vidjeti na Slici 36.



Slika 36. Izračun boje (bajta)

```

on start
  RESET PROGRAM
  PAINT display: color 85
  
```

Slika 37. Primjer programa za PAINT funkciju

3.9. Prikaz crno/bijelo ili bijelo/crno - SCREEN MODE

Ekran može postavljen u "normalan" mod (0) - bijeli ekran s crnim ispisom, ili u **inverzni** (obrnuti) mod (1) - crni ekran s bijelim ispisom. Standardno (default) je ekran postavljen u "normalan" mod (0). Promjenom moda putem funkcije **SCREEN MODE** mijenja se kompletan sadržaj ekrana. Isprobajte slijedeći primjer na Slici 37. Možete ga dopuniti tekstom.

```

forever
  SCREEN MODE 0
  pause (ms) 500
  SCREEN MODE 1
  pause (ms) 500

on start
  RESET PROGRAM
  
```

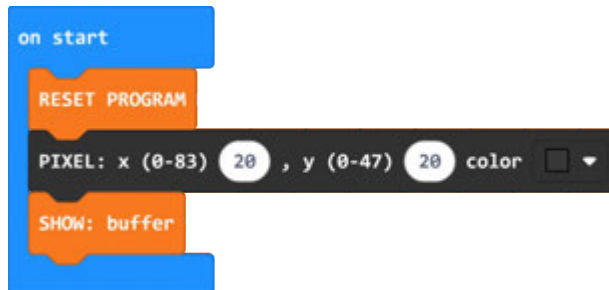
Slika 38. Primjer programa za PAINT funkciju

3.10. Ispis piksela



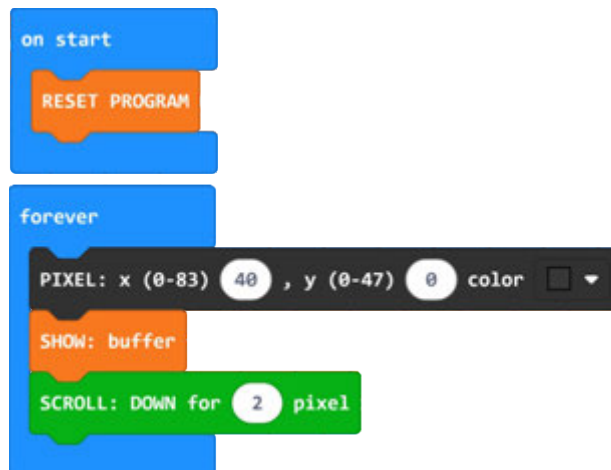
Slika 39.

Funkcija za ispis piksela na ekan u grafičkom modu.

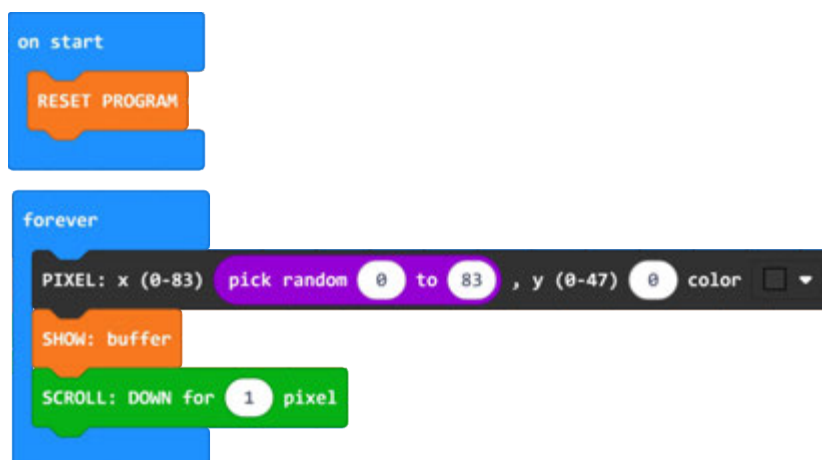


Slika 40.

Ispis jednog piksela na ekan u grafičkom modu, na poziciji x=20, y=20.



Slika 41.



Slika 42.

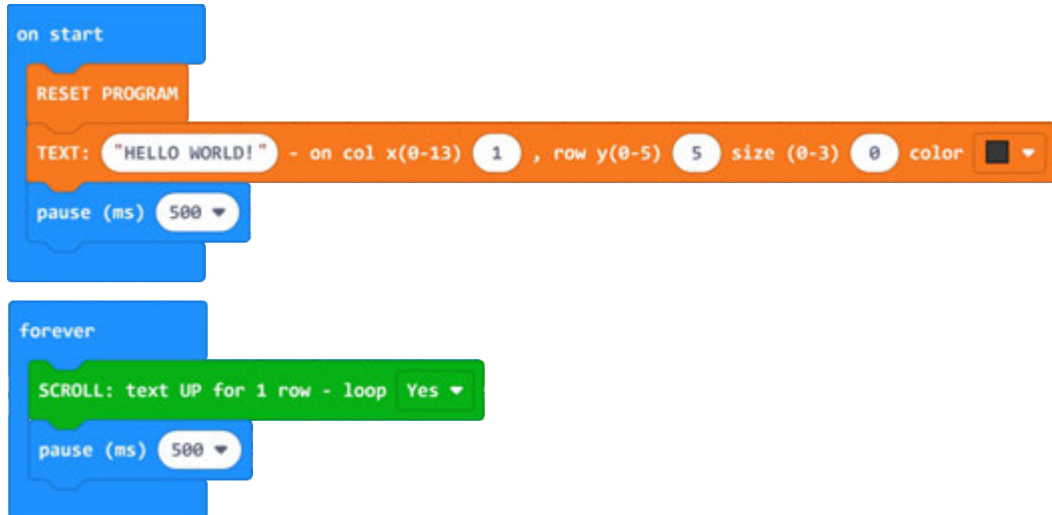
Isprobajte programe na prethodnim slikama 41 i 42.

4. PROGRAMIRANJE - FUNKCIJE POMAKA - SCROLL

4.1. Pomak teksta prema GORE za jednu liniju

SCROLL: text UP for 1 row - loop Yes ▾

Slika 43. Funkcija za pomak (SCROLL) teksta prema GORE



Slika 44. Primjer vertikalnog pomaka teksta

Tekst ispisan na početku pomiče se za jednu liniju prema gore, svakih pola sekunde (Slika 44). Pomak teksta može se koristiti za sve veličine ispisa.

Funkcija LOOP ima dva stanja YES i NO. Probajte promijeniti stanje u NO.
Koja je razlika u pomaku teksta prema prethodnom stanju (YES)?

4.2. Pomak teksta prema DOLJE za jednu liniju

SCROLL: text DOWN for 1 row - loop Yes ▾

Slika 45. Funkcija za pomak (SCROLL) teksta prema DOLJE



Slika 46. Primjer vertikalnog pomaka teksta

Tekst ispisan na početku pomiče se za jednu liniju prema dolje, svakih pola sekunde (Slika 46). Pomak teksta može se koristiti za sve veličine ispisa.

Funkcija LOOP ima dva stanja YES i NO. Probajte promijeniti stanje u NO.
Koja je razlika u pomaku teksta prema prethodnom stanju (YES)?

4.3. Pomak ekrana (slike) prema GORE za jedan ili više piksela (liniju piksela)



Slika 47. Funkcija za pomak (SCROLL) ekrana prema GORE



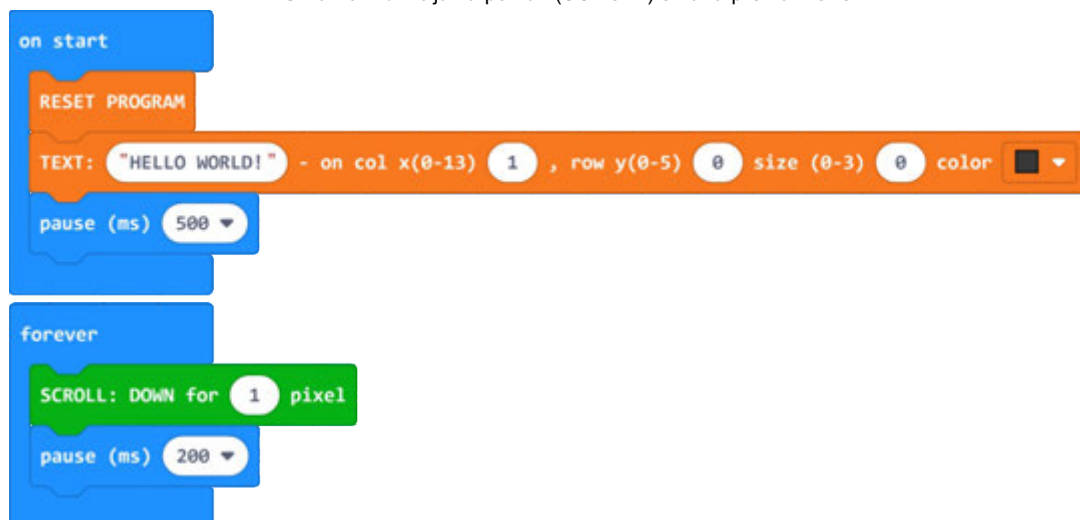
Slika 48. Primjer pomaka ekrana za jedan piksel prema GORE

Tekst ispisan na početku pomiče se za jednu liniju piksela prema GORE, svakih 200 mili sekundi (Slika 48). Pomak ekrana (slike) se može povećati upisom većeg broja u polje **for**.

4.4. Pomak ekrana (slike) prema DOLJE za jedan ili više piksela (liniju piksela)



Slika 49. Funkcija za pomak (SCROLL) ekrana prema DOLJE



Slika 50. Primjer pomaka ekrana za jedan piksel prema GORE

Tekst ispisan na početku pomiče se za jednu liniju piksela prema DOLJE, svakih 200 mili sekundi (Slika 50). Pomak ekrana (slike) se može povećati upisom većeg broja u polje **for**.

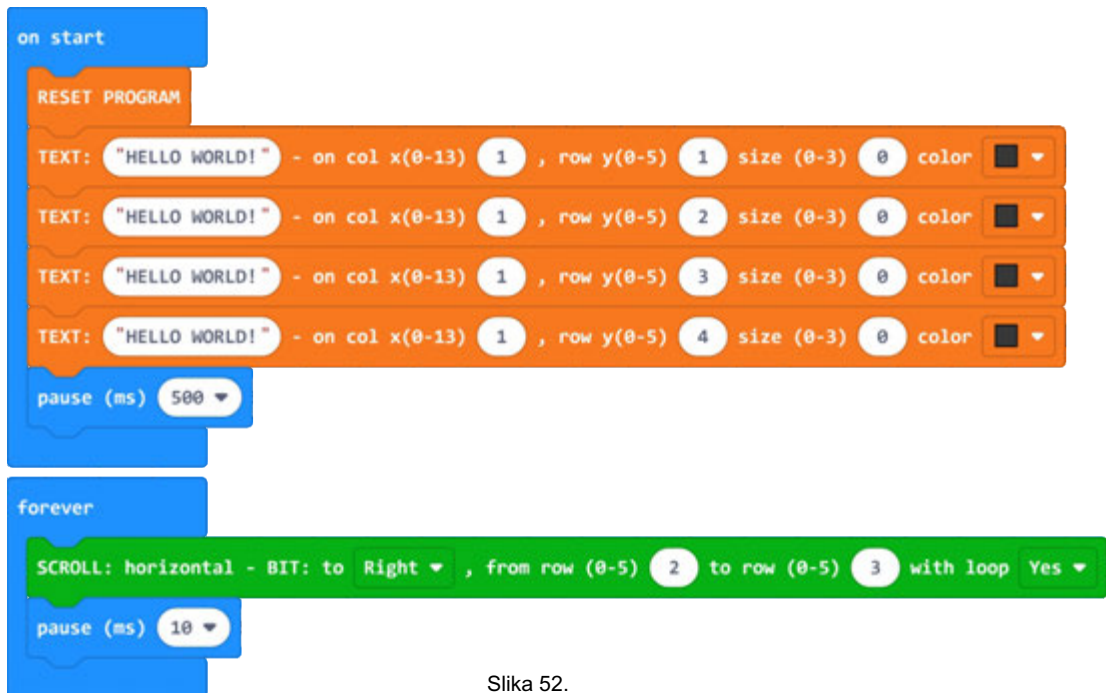
4.5. Horizontalni pomak ekrana (slike) za jedan piksel



Slika 51.

Funkcija za horizontalni pomak ekrana (slike) za jedan piksel. Funkcija omogućava odabir smijera pomaka (LIJEVO - left ili DESNO - right), područja ekrana od linije (tekst) do linije ili cijelog ekrana (0-5).

Kao i kod pomaka teksta može se uključiti opcija LOOP za kružni pomak teksta ili slike.



Slika 52.

Primjer pomaka slike u desnu stranu (Right) dvije srednje linije teksta s kružnim prikazom (Slika 52). Dodajte redove za ispis teksta u liniji (row) 0 i 5, te promijenite u funkciji SCROLL vrijednost 2 u 0 i 3 u 5.

Koja se promjena desila?

Probajte promijeniti smijer pomaka.

4.5. Animacija funkcijama za pomak

```

on start
  RESET PROGRAM
  LINE: x1(0-83) 32 y1(0-47) 40 to x2(0-83) 52 y2(0-47) 40 , color 
  LINE: x1(0-83) 32 y1(0-47) 41 to x2(0-83) 52 y2(0-47) 41 , color 
  CIRCLE: x(0-83) 20 y(0-47) 20 , radius 15 , color  filled with 
  CIRCLE: x(0-83) 62 y(0-47) 20 , radius 15 , color  filled with 
  set index to 11
  repeat 4 times
    do
      CIRCLE: x(0-83) 20 y(0-47) 20 , radius index , color  filled with null 
      CIRCLE: x(0-83) 62 y(0-47) 20 , radius index , color  filled with null 
      change index by 1
  repeat 5 times
    do
      SCROLL: horizontal - BIT: to Left , from row (0-5) 0 to row (0-5) 5 with loop Yes 
      pause (ms) 10
  forever
    repeat 10 times
      do
        SCROLL: horizontal - BIT: to Right , from row (0-5) 0 to row (0-5) 5 with loop Yes 
        pause (ms) 10
      repeat 10 times
        do
          SCROLL: horizontal - BIT: to Left , from row (0-5) 0 to row (0-5) 5 with loop Yes 
          pause (ms) 10
  
```

Slika 53.

Jednostavna animacija putem različitih funkcija.

Isprobajte!

5. PROGRAMIRANJE - GAME FUNKCIJE

5.1. Kontrola LED svjetla

Sučelje ima ugrađena dva LED svjetla, jedno CRVENO i jedno ZELENO. CRVENO se nalazi ispod ekrana na lijevoj strani, a ZELENO na desnoj strani.



Slika 56.



Slika 57.

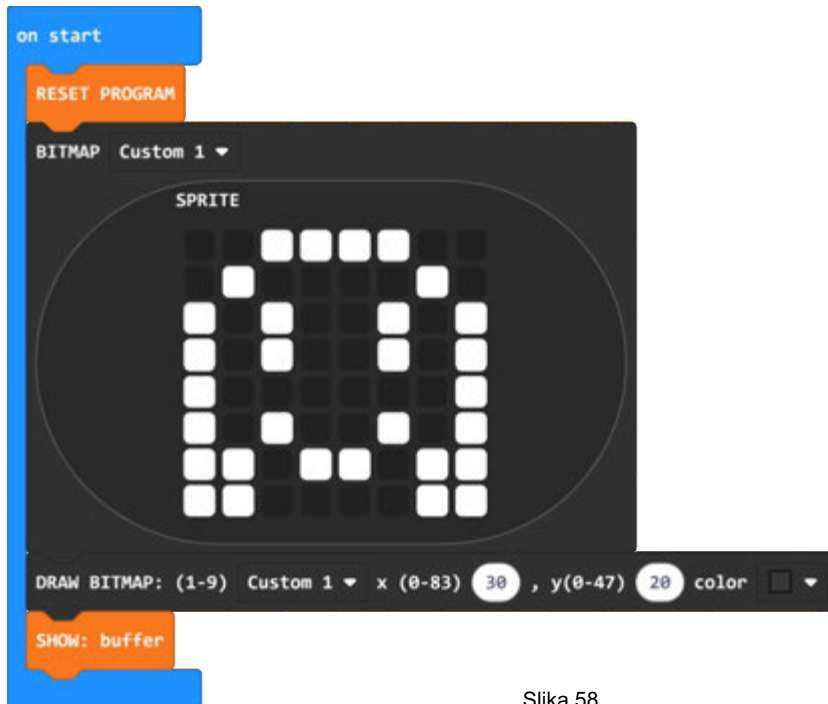
Primjer programa koji naizmjenično pali CRVENO i ZELENO led svjetlo (Slika 57).

5.2. Kreiranje BITMAPE (sprajta) - objekata

Kreiranje grafičkih objekata (BIT-mapa ili sprajtova) izvodi se u grafičkom modu (bufferu - memoriji) radi bržeg ispisa na ekranu i izbjegavanja određenih loših efekata (titranja). Zato se prvo objekt (jedan ili više objekata) spremaju u po moćnu memoriju (buffer), a na kraju se memorija putem funkcije SHAW buffer prikazuje na ekranu.

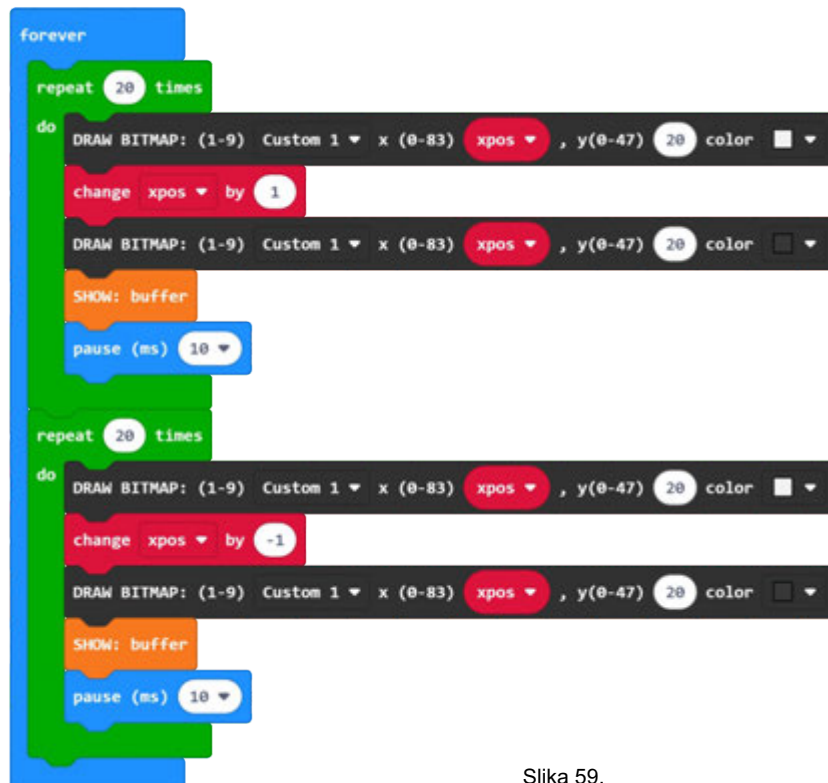
Nakon kreiranja objekta (BITMAP), u ovom primjeru CUSTOM 1, potrebno je odrediti poziciju na kojoj će se objekt iscrtati i kojom BOJOM (DRAW BITMAP).

Kratki primjer s osnovnim funkcijama prikazan je na Slici 58.



Slika 58.

Boja ispisa nam omogućava da objekt ispišemo CRNOM bojom, a obrišemo BIJELOM bojom.

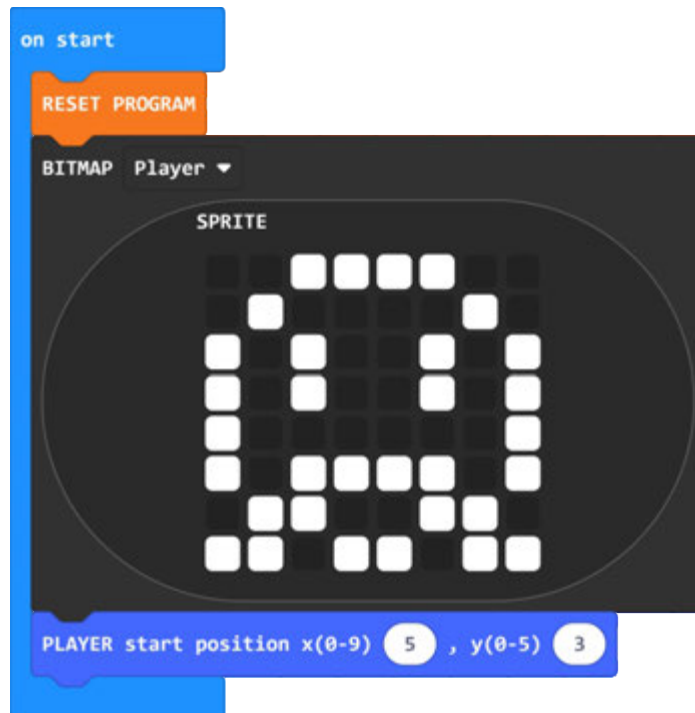


Slika 59.

Dopunite prethodni program prikazan na Slici 58 blokom **forever** prikaznom na Slici 59.

5.3. Objekat igrača (PLAYER)

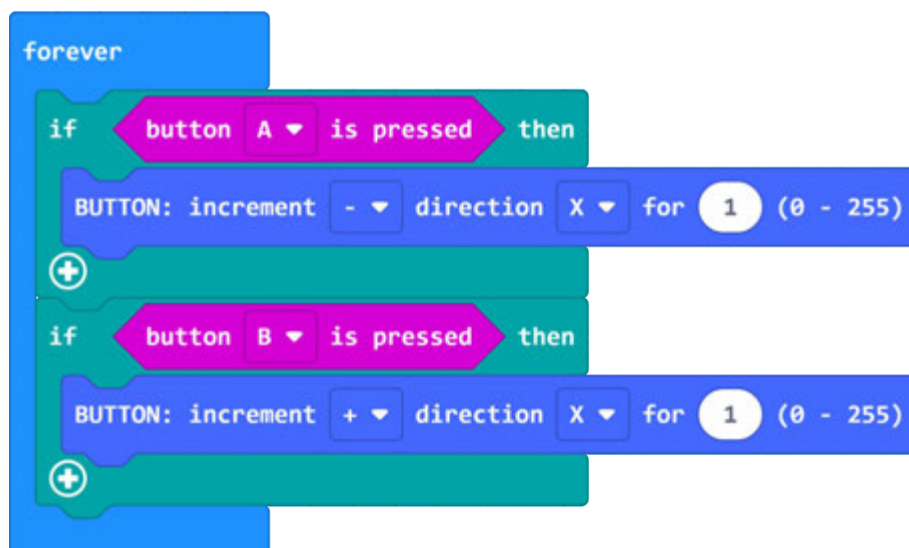
Ako želimo da BITMAPA bude objekt igrača u izborniku treba odabrati **Player**. Nakon kreiranja objekta potrebno je pokrenuti funkciju za njegov prikaz na ekranu. Za pozicioniranje se koristi tekst mod rezolucija (14 x 6). U slijedećem primjeru, objekt igrača se iscrtava na poziciji x=5, y=3.



Slika 61.

Funkcija PLAYER ujedno prikazuje stanje pomoćne memorije (BUFFER) na ekranu, pa nije potrebno pozivanje funkcije **SHOW BUFFER**.

5.4. Horizontalne kontrole pomaka



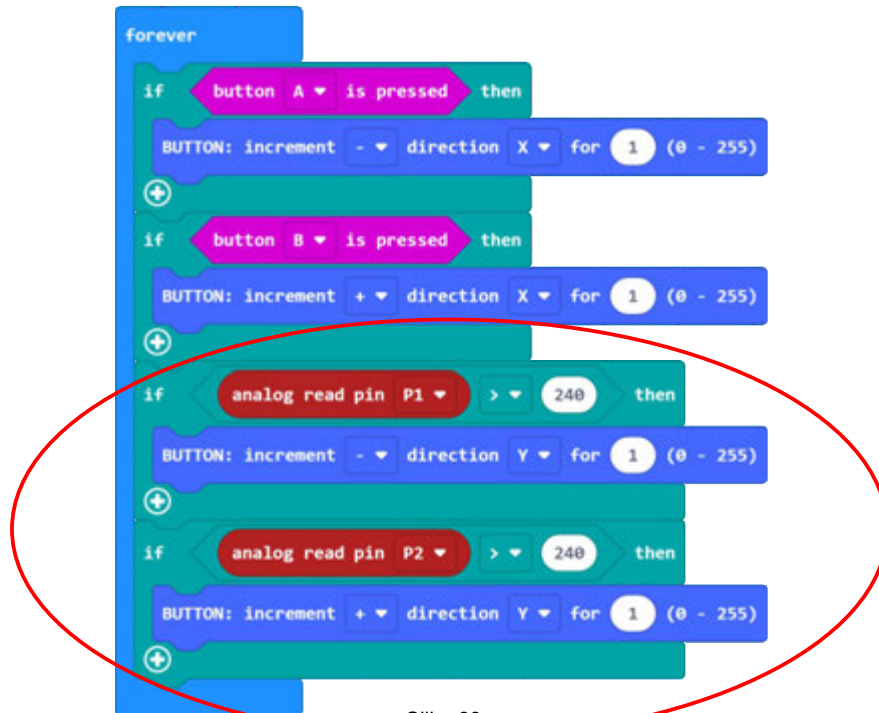
Slika 62.

U prethodni program dodajte blok **forever** prikazan na Slici 62.

Isprobajte kako radi program ako funkciji **BUTTON** promijenite vrijednosti za smjer (**direction**) i **for** (piksel).

5.5. Vertikalne kontrole (dodirni senzori)

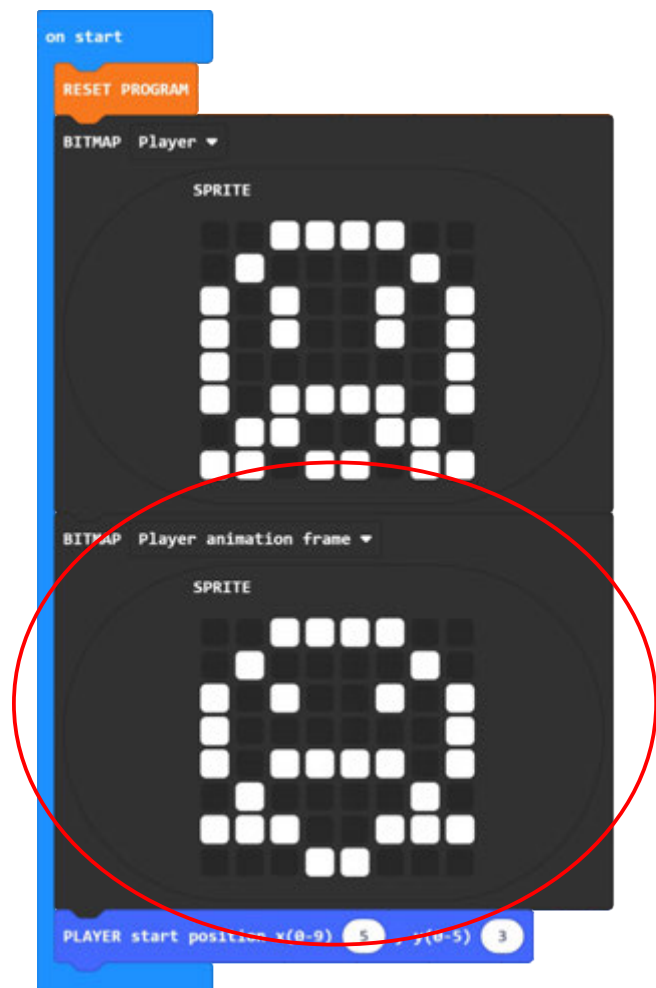
Za upravljanje objektom igrača u svim smjerovima potrebno je dodati i kontrolu dodirnih senzora koji se nalaze na donjoj strani micro:bita. Analogno očitavanje dodirnih senzora nije jednako sa priključenim USB kablom na micro:bit i kad nije priključen. Primjer na Slici 63 prikazuje vrijednosti sa priključenim USB kablom. Vrijednost bez priključenog USB kabla na micro:bit je < 100 .



Slika 63.

5.6. Animacija objekta igrača

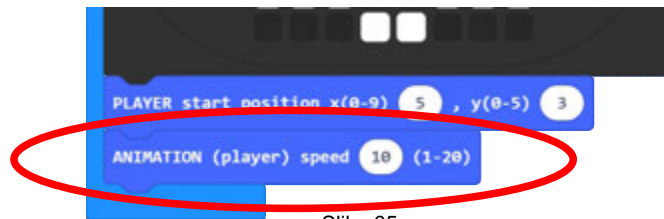
Za animaciju objekta igrača potrebna je još jedna bitmapa. Dopunite program prema Slici 64. Kretanjem objekta igrača na ekranu se naizmjenično iscrtavaju bitmape **Player** i **Player animation frame**.



Slika 64.

5.7. Kontrola brzine animacije

Animacija objekta igrača može biti brža ili sporija. Za kontrolu brzine koristimo funkciju **ANIMATION (player) speed**. Ako želimo da animacija (izmjena bitmapa) bude sporija potrebno je upisati veću vrijednost u polje **speed**. **Animacija se izvodi samo dok je objekt igrača u pokretu.**



Slika 65.

Dopunite startni blok programa funkcijom za kontrolu brzine animacije prema Slici 65. Probajte sa različitim vrijednostima za brzinu.

5.8. START game

Imamo osnovnu konstrukciju igre sa kontrolom kretanja objekta igrača. Na početku trebamo staviti poruku koja će se ispisati nakon pokretanja programa. Funkciju za prikaz standardne poruke (**GAME: START message**) umetnite u blok start kao što je prikazano na Slici 66. Nakon ispisa potrebno je pokrenuti funkciju **pause**, tako da se poruka stigne pročitati, a nakon toga obrisati ekran funkcijom **CLEAR SCREEN**. Ako ne pokrenemo funkciju za brisanje ekrana, objekt igrača će se iscrutati preko startnog teksta.



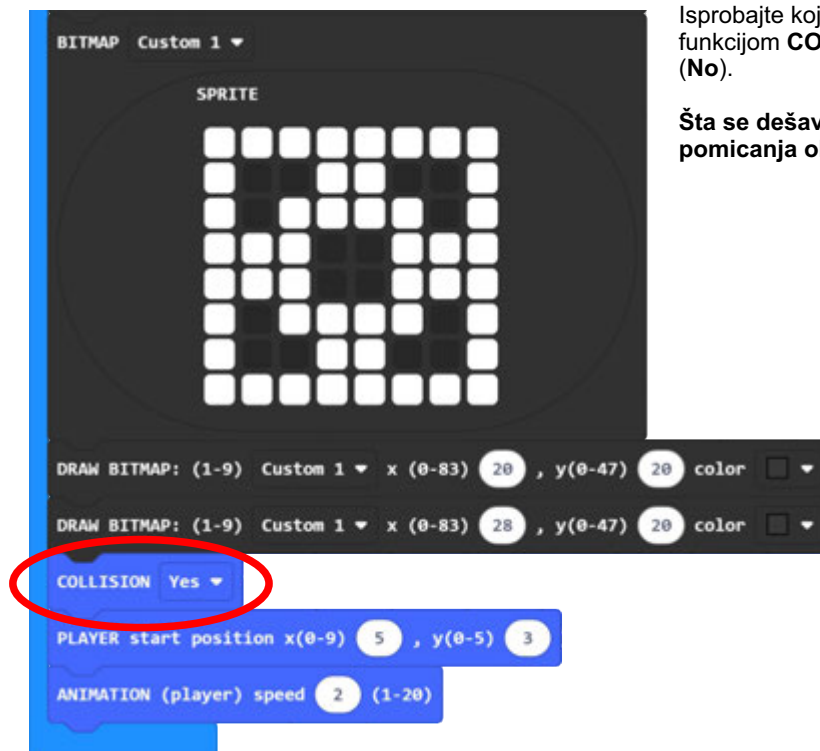
Slika 66.

5.9. COLLISION funkcija

Dopunite program novim objektom (**Custom 1**). Iscrtajte ga na ekran (**DRAW BITMAP**) nekoliko puta, na različite pozicije prema Slici 67 ili proizvoljno. Dopunite program funkcijom **COLLISION**.

Isprobajte koja je razlika sa uključenom funkcijom **COLLISION (Yes)** i isključenom (**No**).

Šta se dešava sa objektima prilikom pomicanja objekta igrača prema njima?



Slika 67.

5.10. Gravitacija - GRAVITY

Da bi kretanje objekta igrača bilo što prirodnije, te da bi mogao skakati i padati potrebno je u igru uključiti funkciju gravitacije - GRAVITY (Slika 68 zaokruženo).

Radi dodavanja funkcije GRAVITY (gravitacija) za kontrolu objekta igrača koristimo samo horizontalne kontrole. Vertikalne smo maknuli iz programa.

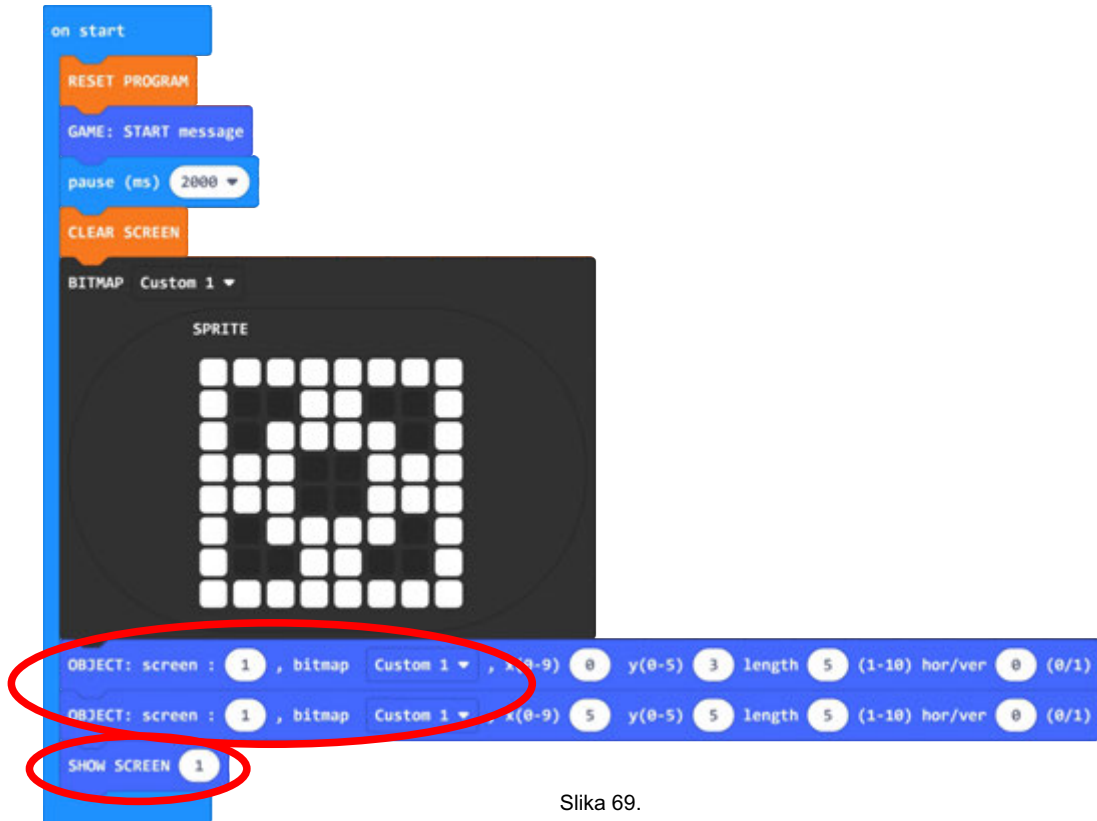
Kompletan program je prikazan na Slici 68.

Koja je razlika u kretanju objekta igrača sa uključenom gravitacijom i isključenom?

Slika 68.

5.11. Kreiranje objekata (horizontalnih i vertikalnih) - max. 20 objekata

Objekti koji su duži od jedne bitmape (8x8 piksela) mogu biti položeni horizontalno ili vertikalno. Objekti se kreiraju višekratnim ponavljanjem iste bitmape. **Objekti putem kojih se zarađuju bodovi ili gube životi** običajeno su dužine jedne bitmape, ako su duži, samo prva pozicija je aktivna za dobivanje bodova ili gubitak života.

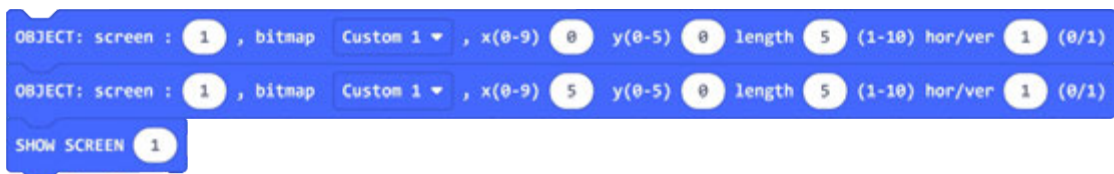


Slika 69.

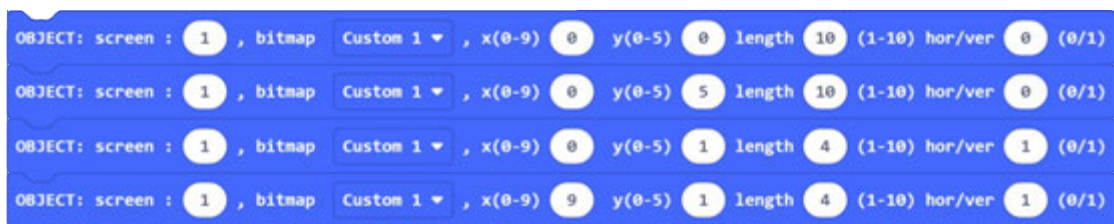
Uzeli smo dio prethodnog programa (Slika 68) i dodali naredbe za kreiranje objekata (Slika 69). Prvo je potrebno definirati objekte koji će se iscrtati na ekranu putem naredbe OBJECT. Sve naredbe se grupiraju u «ekrane» koji se putem naredbe SHOW SCREEN prikazuju na ekranu. U ovom primjeru definiramo «ekran» 1 s dva objekta. Oba objekta su složena od iste bitmape (custom1). Za pozicioniranje potrebno je odrediti x i y poziciju početne bitmape objekta. Dužina (broj ponavljanja) se određuje upisom vrijednosti u polje **length**. Maksimalna horizontalna dužina je 11 ($84 / 8 = 10,5$ bitmapa).

Za vertikalno iscrtavanje bitmape potrebno je promijeniti vrijednost polja «hor/ver» u 1.

U prethodnom programu (Slika 69) napravite izmjene vrijednosti prema Slici 70. Isprobajte program.



Slika 70.



Slika 71.

Isprobajte kombinaciju naredbi prema Slici 71. Ne zaboravite naredbu **SHOW SCREEN (1)** koja dolazi na kraju.

5.12. Kreiranje više od jednog «ekrana» - max. 5 «ekrana»

Ako želimo kreirati više različitih «ekrana» potrebno je za svaki «ekran» kreirati objekte. Slijedeći primjer je sa dva «ekrana» i tri objekta.

		(ekran 1)										(ekran 2)									
		x										x									
		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
y	0																				
	1																				
	2											A	A	A	A	A					
	3																	B	B	B	B
	4	X	X	X	X	X	X	X	X	X	X										
5																					

Slika 72.

Kada želite kreirati više «ekrana» dobro je napraviti skicu kao što prikazuje primjer na Slici 72. Ekranе možemo skicirati putem tabele u Excelu ili raster papiru. Na taj način je puno lakše vizualizirati sve ekrane, naročito ako se koriste horzontalni i vertikalni objekti.

```

on start
  RESET PROGRAM
  GAME: START message
  pause (ms) 2000
  CLEAR SCREEN
  BITMAP Custom 1
  SPRITE
  OBJECT: screen : 1 , bitmap Custom 1 , x(0-9) 0 y(0-5) 4 length 10 (1-10) hor/ver 0 (0/1)
  OBJECT: screen : 2 , bitmap Custom 1 , x(0-9) 0 y(0-5) 2 length 5 (1-10) hor/ver 0 (0/1)
  OBJECT: screen : 2 , bitmap Custom 1 , x(0-9) 6 y(0-5) 3 length 4 (1-10) hor/ver 0 (0/1)
  SHOW SCREEN 1

on button B pressed
  CLEAR SCREEN
  SHOW SCREEN 1

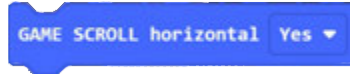
on button A pressed
  CLEAR SCREEN
  SHOW SCREEN 2
    
```

Slika 73.

Primjer na Slici 73 koristi dva «ekrana» koji se mogu putem tipke A ili B naizmjenično prikazati.

5.13. Tekući «ekrani» - prikaz ekrana s horizontalnim pomakom (scroll)

Kod kreiranja platformске igre koristimo pokretne platforme koje se pomiču s jedne strane ekrana u drugu. Da bi pokrenuli platforme (objekte) koje smo kreirali u prethodnom programu (Slika 73) trebamo dodati aktiviranu naredbu **GAME SCROLL horizontal (yes)** (Slika 74.). Objekti na «ekranima» se ispisuju na ekranu u kružnom redoslijedu brojeva «ekrana» (1,2,1,2,1,2,...)

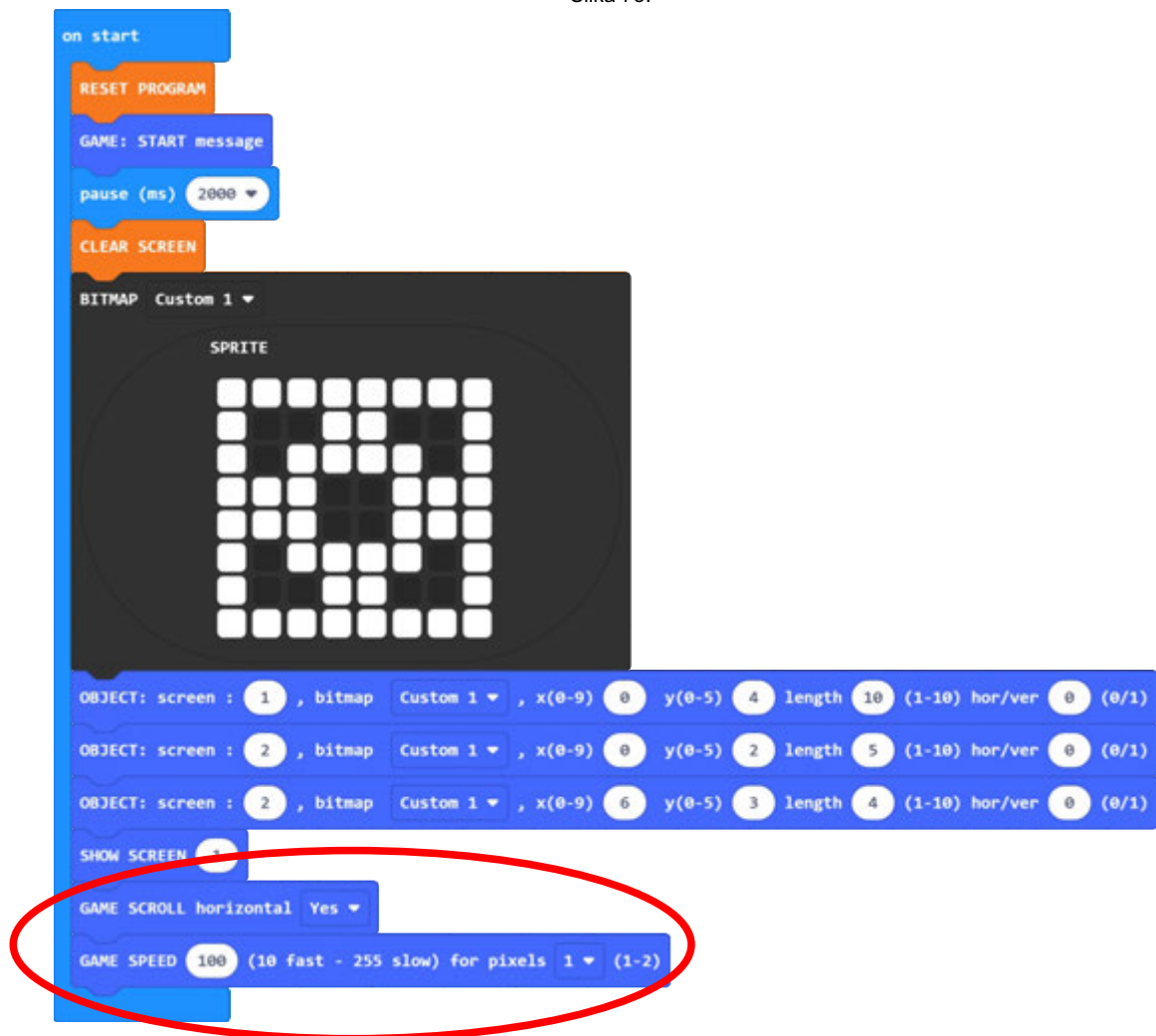


Slika 74.

Brzina horizontalnog pomaka objekata može se mijenjati naredbom **GAME SPEED** (Slika 75.). Manja vrijednost varijable znači veću brzinu pomaka. Maksimalna brzina je ograničena na **10**, a početna (default) je postavljena na 100 (0 = 10). Brzinu možemo povećati i pomakom za 2 piksela (default 1), ali pomak više neće biti tako 'fini' kao kod pomaka za 1 piksel. Ako program sadrži **puno objekata** za kontrolu, prilikom maksimalne brzine (10) se može desiti da prestane s radom. **U tom slučaju trebate smanjiti brzinu, jer program ne stigne obraditi sve operacije u prekratkom vremenu.**



Slika 75.



Slika 76.

Da bi vidjeli koje su razlike u brzini pomaka probajte program s različitim vrijednostima za brzinu. Možete proširiti program za još jedan «ekran» (3).

5.14. Pokretanje objekta igrača (player) kod ekrana s pomakom (scroll)

Skok 'igrača' direktno prema gore kontroliramo putem naredbe **JUMP UP**, koja se najčešće koristi kod platformske igre gdje se igrač nalazi istoj poziciji na ekranu (horizontalno). Visina skoka se određuje u pikselima. Da bi funkcija bila aktivna morali smo dodati još neke obavezne funkcije za poziciju i kontrolu kretanja objekta **igrača** (COLLISION, GRAVITY, PLAYER start position).

JUMP UP: pixel 0 (0-255)

Slika 77.

on start

RESET PROGRAM

GAME: START message

pause (ms) 2000

CLEAR SCREEN

BITMAP Player

SPRITE

BITMAP Custom 1

SPRITE

OBJECT: screen : 1 , bitmap Custom 1 , x(0-9) 0 y(0-5) 4 length 10 (1-10) hor/ver 0 (0/1)

OBJECT: screen : 2 , bitmap Custom 1 , x(0-9) 0 y(0-5) 2 length 5 (1-10) hor/ver 0 (0/1)

OBJECT: screen : 2 , bitmap Custom 1 , x(0-9) 6 y(0-5) 3 length 4 (1-10) hor/ver 0 (0/1)

SHOW SCREEN 1

GAME SCROLL horizontal Yes

GAME SPEED 30 (10 fast - 255 slow) for pixels 1 (1-2)

COLLISION Yes

GRAVITY Yes

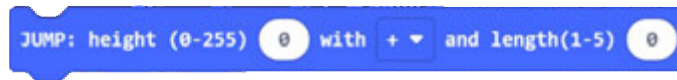
PLAYER start position x(0-9) 4 , y(0-5) 0

on button A pressed

JUMP UP: pixel 15 (0-255)

Slika 78.

Za skok **igrača** u desnu (+) ili lijevu (-) stranu koristimo naredbu **JUMP**. Osim visine skoka, kao i kod naredbe JUMP UP, određujemo u koju stranu će se **igrač** pri skoku kretati **+** = **desnu** ili **-** = **lijevu**. Upisom vrijednosti od 0 do 5 određujemo kut skoka. Za okomiti skok (na mjestu) vrijednost je **0**, za skok pod kutom od 45 stupnjeva vrijednost (**length**) je 1. Vrijednosti 2 - 5 povećavaju kut (**length**) skoka.



Slika 79.

Prethodnom programu (Slika 78.) izmijenite funkciju button A i dodajte funkciju button B prema doljnoj slici (Slika 80).



Slika 80.

5.15. Kontrolne funkcije

5.15.1. Status igre - GET GAME status

Da bi program koji se pokreće u micro:bitu mogao izvršiti neke funkcije potrebno je očitati određene vrijednosti koje se koriste u igri. Status igre se koristi za izvođenje zvučnih efekata tijekom igre i da bi se znalo kada je kraj igre. Funkcija se poziva **SAMO JEDNOM** na početku **forever** petlje.



Slika 81.

5.15.2. Zvučni i svijetlosni efekti - GAME: all sounds.

Za izvođenje određenih zvučnih i svijetlosnih efekata povezanih uz određeni događaj u igri (bod, gubitak života, pad) koristimo funkciju **GAME: all sounds**. Obavezno je prethodno pokrenuti funkciju **GET GAME status**.



Slika 82.

5.15.3. Poruka za kraj igre - GAME: END message.

Da bi program završio igru porukom, potrebno je uključiti funkciju **GAME: END message**. Obavezno je prethodno pokrenuti funkciju **GET GAME status**.



Slika 83.

5.15.4. Bodovi - POINTS at start

Za praćenje broja osvojenih bodova u igri potrebno je pokrenuti funkciju **POINTS at start** sa početnim brojem bodova. U igri u kojoj samo možemo osvojiti plus bodove običajeno je početna vrijednost 0. U igri sa mogućnosti dobitka i gubitka bodova, početna vrijednost je veća od nule. Da bi **igrač** dobio bodove potrebno je u igru uključiti **Points (+)** objekte.



Slika 84.

5.15.5. Životi - LIVES at start

Za gubitak života, u igri, potrebno je pokrenuti funkciju **LIVES at start** koja postavlja početnu vrijednost broja života u igri. Gubitak života se događa prilikom pada (FALL) ili dodira sa objektom **Lives (-)**.



Slika 85.

5.15.6. Pad gubitak života - FALL

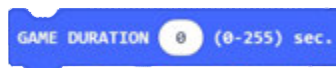
Igra se sastoji od platformi kojima se **igrač** kreće. Prilikom pada objekt igrača može izgubiti život ili se samo ponovo pojaviti na početnoj poziciji. Uključivanjem funkcije **FALL** uključujemo gubitak života prilikom pada van ekrana.



Slika 86.

5.15.7. Ograničeno trajanje igre - GAME DURATION

Za definiranje vremena trajanja igre u sekundama potrebno je uključiti funkciju **GAME DURATION**. Funkciju koristimo u igrama u kojima je cilj skupiti što više bodova u jednakom vremenskom roku. U takvim igrama se koristi funkcija samo za bodove.



Slika 87.

5.15.8. Negativni bodovi - POINTS negative

Ako želite uključiti oduzimanje bodova potrebno je uključiti funkciju **POINTS negative**. Da bi ova funkcija bila aktivna potrebno je aktivirati i funkciju **LIVES at start**.



Slika 88.

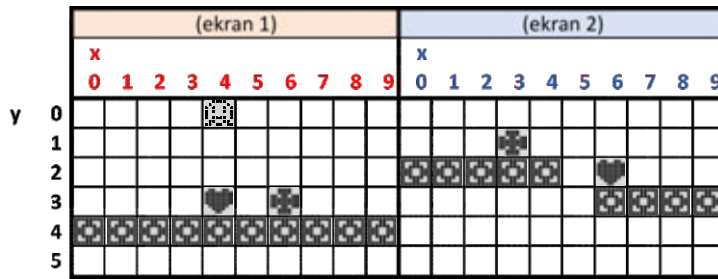
5.15.9. Redosljed prikaza «ekrana» slučajnim odabirom - RANDOM displays flow

Ako igra ima **VIŠE OD DVA «ekrana»**, da bi izbjegli stalno ponavljanje istog redosljeda možemo uključiti ovu funkciju. Funkcija slučajnim odabirom (**random**) kreira redosljed prikazivanja «ekrana».



Slika 89.

5.16. Kompletna platformska igra



Slika 90.

Da bi igra bila kompletna dodali smo u svaki «ekran» objekte za dobivanje bodova (♥) i gubitak života (☹) prema pozicijam na skici (Slika 90.), te objekt za animaciju igrača.

```

on start
  RESET PROGRAM
  GAME: START message
  pause (ms) 2000
  CLEAR SCREEN

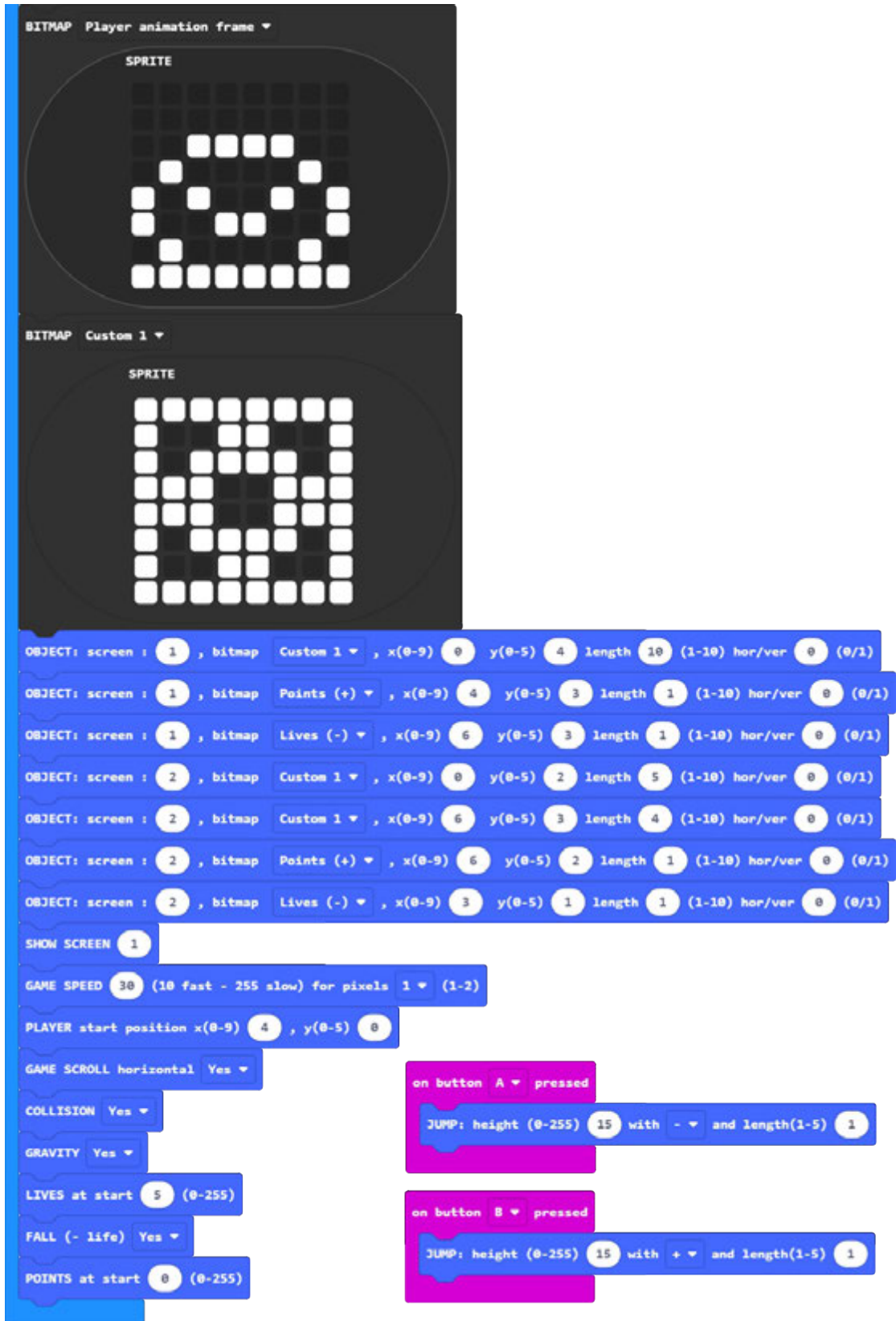
  BITMAP Points (+)
  SPRITE

  BITMAP Lives (-)
  SPRITE

  BITMAP Player
  SPRITE

  forever
    GET GAME status
    GAME: all sounds
    GAME: END message
  
```

(nastavak programa na sljedećoj stranici)



Slika 91.

5.16.1. Brisanje podataka iz memorije - DELETE past objects

Tijekom izrade programa i mjenjanjem definicija objekata može se desiti, da neki objekti koje ste izbrisali iz programa, ostanu zapisani u memoriji sučelja. U tom slučaju bi se mogli na ekranu pojaviti **'fantomski'** objekti koje ste obrisali iz programa. Da bi se to izbjeglo na početku programa možete dodati funkciju **DELETE past objects**.

Ova funkcija se može maknuti iz programa, nakon što je program završen.



Slika 92.

5.16.2. Automatska kontrola nivoa (levela) igre - AUTO LEVELS

Da bi igra bila zahtjevnija možemo dodati više težinskih novoa. Viši nivo ima veću brzinu izvođenja igre i samim time ga je teže završiti. Uz funkciju za automatsku kontrolu možemo odrediti vrijednosti koje određuju nivoe igre. Na početku (**speed max.**) upisujemo vrijednost koja definira najveću brzinu igre (zadnji nivo). Nakon toga početnu brzinu (**speed start**) kojom započinjemo igru. Za koliko se povećava brzina igre prelaskom na viši nivo upisuje se u polje **change for**. Zadnja vrijednost (**points for new level**) određuje koliko je potrebno bodova osvojiti da bi se prešlo na viši nivo.

Ova funkcija možda neće podržavati sve oblike igara.

Iz prethodnog programa obrišite **GAME SPEED** funkciju i dodajte **AUTO LEVELS** s vrijednostima iz primjera na slici 93.



Slika 93.

5.16.3. Brzina razmjene podataka (micro:bit - AD display) - SET COM FACTOR

Na početku rada programa, kad se šalje puno podataka za definiranje različitih funkcija i objekata potrebno je **COM FACTOR** postaviti na **8** (početna vrijednost) ili više. Na taj način program AD sučelja ima dovoljno vremena da obradi sve podatke. U koliko je vrijeme prekratko (brzina prevelika), program neće stići obraditi sve podatke koje mu micro:bit pošalje i nedostajati će neki objekti ili neće raditi neke funkcije, ili će program prestati sa radom.

Ako želite da se radnje (nakon dijela programa koji šalje postavke za objekte i funkcije) odvijaju brže, te da igra bude što brža, možete **COM FACTOR** postaviti na **4** (najmanja preporučena vrijednost, za najveću brzinu).

Ne preporuča se vrijednost manja od **4**.

Program AD sučelja dozvoljava da probate i sa manjim vrijednostima.



Slika 94.

6. PRIMJER PROGRAMA

6.1. METEORI

Program uključuje neke od funkcija koje su opisane na stranici 28. Igra ima tri «ekrana» koji se prikazuju slučajnim redosljedom uz pomoć funkcije **RANDOM** (5.15.9.) i ograničena je na **30 sekundi** funkcijom **GAME DURATION** (5.15.7.). Skica rasporeda objekata je prikazana na doljnoj slici (Slika 95.).

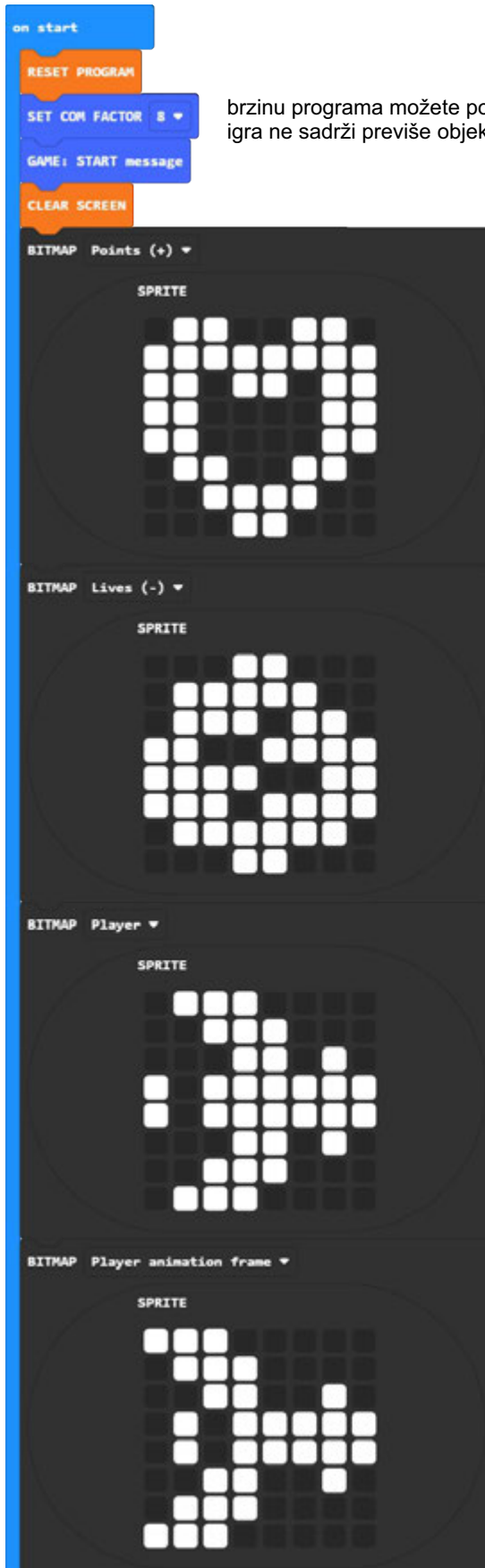
	(ekran 1)										(ekran 2)										(ekran 3)									
	x										x										x									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
y 0																														
y 1																														
y 2																														
y 3																														
y 4																														
y 5																														

Slika 95.

Ako želite otežati osvajanje bodova možete još dodati objekte koji će samo otežavati kretanje **igrača**, kao što je prikazano na Slici 96.

	(ekran 1)										(ekran 2)										(ekran 3)									
	x										x										x									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
y 0																														
y 1																														
y 2																														
y 3																														
y 4																														
y 5																														

Slika 96.



brzinu programa možete povećati ako postavite **COM FAKTOR** na 6 ili 4, ako igra ne sadrži previše objekata. (isprobajte koja brzina je pogodna za vaš program)

(nastavak programa na sljedećoj stranici)

The image shows a sequence of MakeCode blocks for a game. The first part consists of 12 'OBJECT: screen (1-5):' blocks, each with different parameters for bitmap, x, y, length, and hor/ver. Below these are several configuration blocks: 'SHOW SCREEN 1' with a comment 'prikaži prvi «ekran»', 'GAME SPEED 30' with a comment 'brzina igre i pomak', 'PLAYER start position x(0-9) 4, y(0-5) 2' with a comment 'početna pozicija objekta igrača', 'GAME SCROLL horizontal Yes', 'COLLISION Yes', 'POINTS at start 0' with a comment 'broj bodova na početku igre = 0', 'LIVES at start 5' with a comment 'broj života na početku igre = 5', 'GAME DURATION 30' with a comment 'vrijeme igre ograničeno na 30 sekundi', and 'RANDOM displays flow' with a comment 'odabir prikaza «ekrana» slučajnim odabirom redosljeda'. The second part is a 'forever' loop containing 'GET GAME status', 'GAME: all sounds', 'GAME: END message', and two 'if' blocks: 'if button A is pressed then' followed by 'BUTTON: increment + direction Y for 2' with a comment 'kontrola igrača po y osi prema dolje (+)', and 'if button B is pressed then' followed by 'BUTTON: increment - direction Y for 2' with a comment 'kontrola igrača po y osi prema gore (-)'.

Slika 97.

7. PRIMJER PROGRAMA

7.1. OSTALE FUNKCIJE

7.1.1. Preuzimanje pozicije igrača - GAME get player position (x)

Putem ove funkcije možete preuzeti vrijednost za horizontalni (x) položaj **igrača** na ekranu. Vrijednost prikazuje grafičku poziciju igrača (0-83). Ovu funkciju možete koristiti u bilo kojoj kombinaciji pri izradi programa.

GAME: get player position (x)

Slika 98.

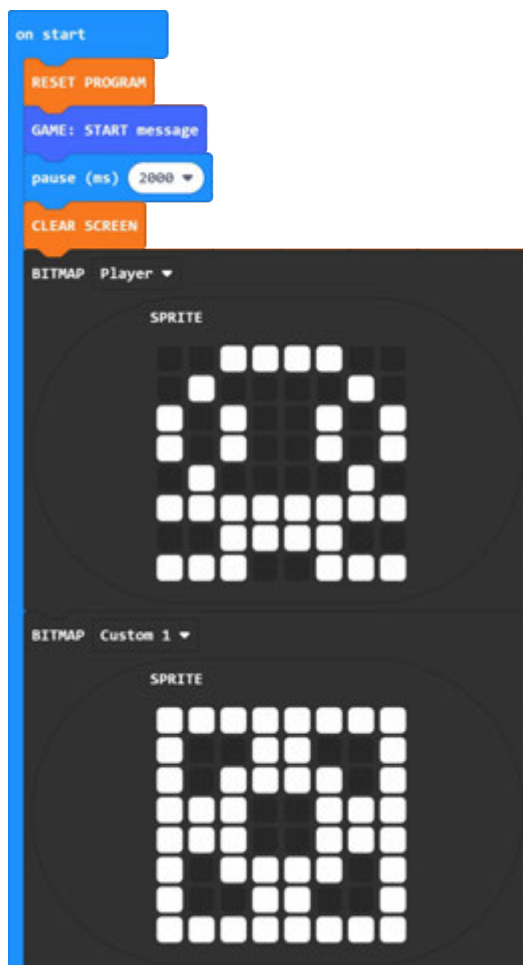
7.1.2. Preuzimanje pozicije igrača - GAME get player position (y)

Putem ove funkcije možete preuzeti vrijednost za vertikalni (y) položaj **igrača** na ekranu. Vrijednost prikazuje grafičku poziciju igrača (0-47). Ovu funkciju možete koristiti u bilo kojoj kombinaciji pri izradi programa.

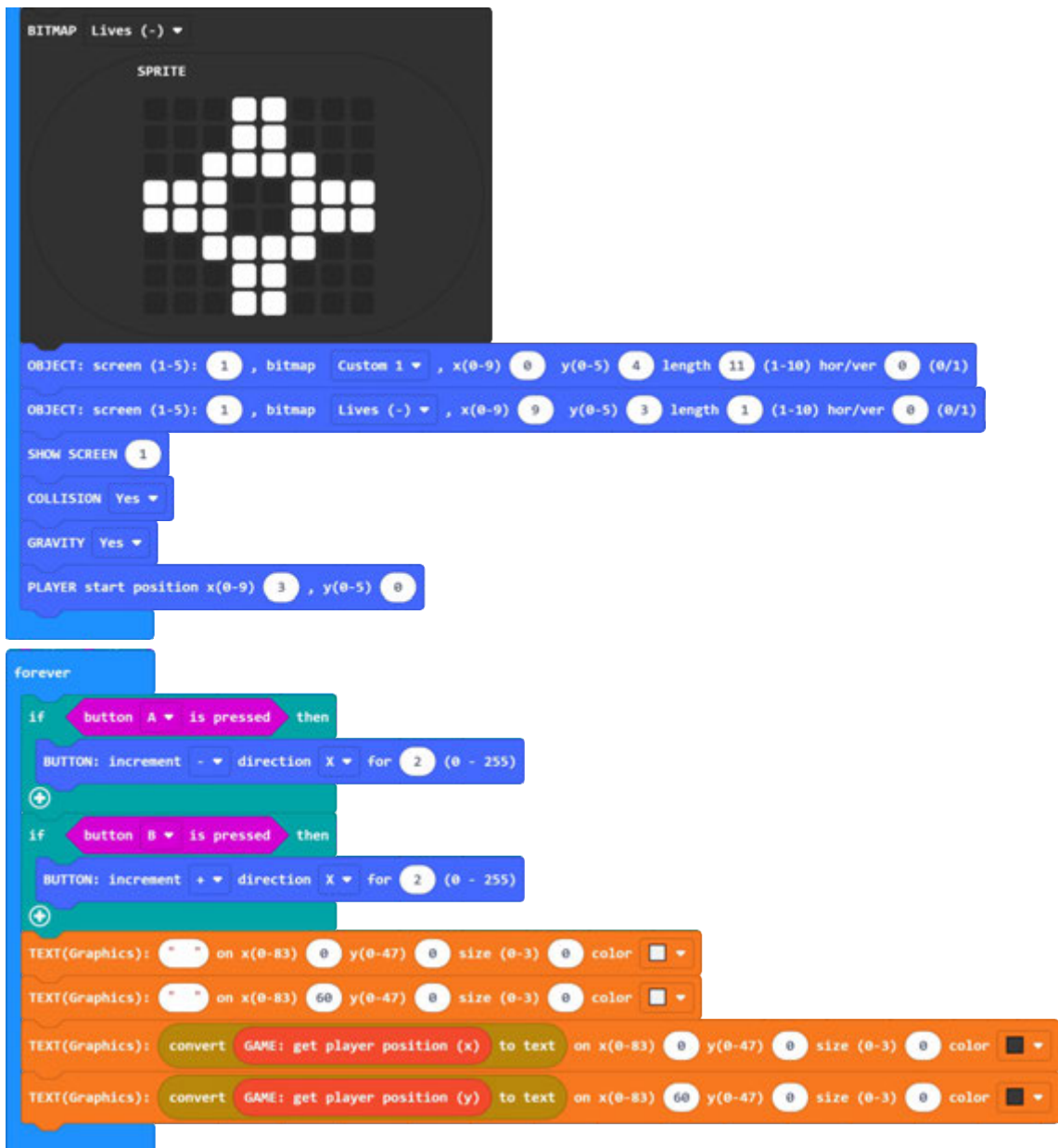
GAME: get player position (y)

Slika 99.

7.2. Primjer programa



(nastavak programa na sljedećoj stranici)



Slika 100.

U ovom primjeru koristimo funkcije za preuzimanje pozicije **igrača** samo radi prikaza na ekranu. Iste funkcije možete koristiti za kontrolu **igrača** ili ograničavanje njegovog kretanja po ekranu.



Slika 101.

Primjer ograničenja horizontalnog kretanja igrača (Slika 101). U prethodnom programu (Slika 100) napravite izmjenu prema primjeru na slici 101.

8. ZAKLJUČAK

Željeli smo napraviti sučelje s ekranom koje će omogućiti prikaz podataka ili izradu jednostavnih igrica. Pri izradi igre, koriste se neke funkcije za definiranje rada igre, a koje imamo i u pravim računalnim igrama (gravitacija). Da bi omogućili maksimalnu kreativnost **većina funkcija nema limitirane vrijednosti**, a to znači da će se dešavati greške kao što je prestanak rada programa ili ispis krivih podataka na ekranu.

Želimo vam ugodan rad.