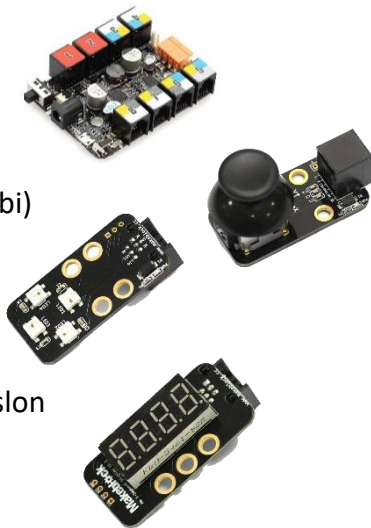


## AKTIVNOST: Igra „Simon“

### Potrebni materijal:

- Orion pločica
- Joystick (ili gumbi)
- LED lampice
- 7-segmentni zaslon
- 3 žice



Simon je klasična igra pamćenja u kojoj je potrebno ponavljati boje koje prikazuje računalo (u našem slučaju Orion pločica). Primjer jedne igre :

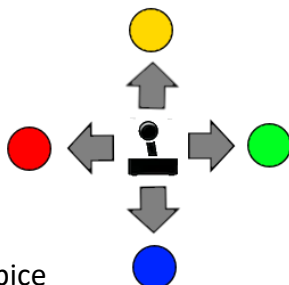
Orion: ●  
Ja: ●  
Orion: ● ●  
Ja: ● ●  
Orion: ● ● ●  
Ja: ● ● ●  
Orion: ● ● ● ●  
Ja: ● ● ● ●  
Pogriješio sam



Napravi igru Simon pomoću Orion pločice i dodataka.

### 1. korak

Napravi program tako da se pomakom Joysticka pale odgovarajuće LED lampice u bojama kao na slici:



### 2. korak

Dodaj Orionu mogućnost da sam pali lampice kao u primjeru igre. Pri tome svaka nova lampica treba biti odabrana slučajno.

### 😊 Savjet:

Budući da Orion treba pamtit i koje je lampice do sada upalio, a to nije jednostavno napraviti pomoću blokova u programu mBlock, možeš napraviti tako da se sve lampice pale u slučajnom redoslijedu u svakoj rundi igre, a ako voliš izazove potraži na internetu *Linear congruential generator*.

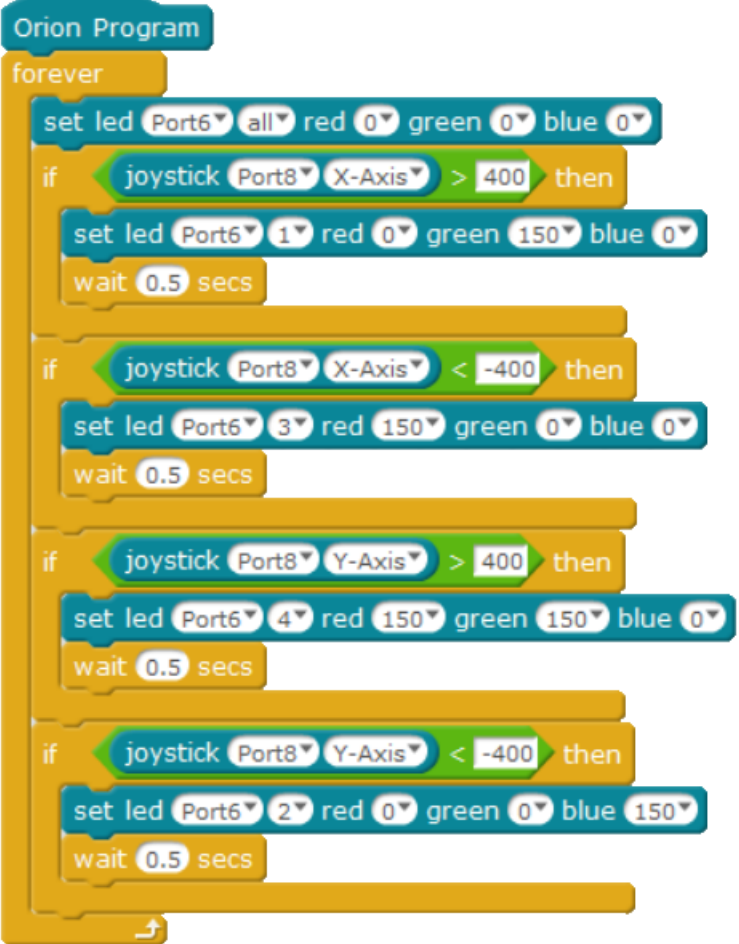
### 3. korak

Provjeri jesu li lampice koje unosi igrač ispravno unesene. Pamti svaki uspješno uneseni niz lampica kao rezultat i prikaži ga na 7-segmentnom zaslonu.

# Rješenje

## 1. korak

Paljenje odgovarajućih LED lampica u određenoj boji pomoću joysticka.

	<p>Prvo ugasimo sve lampice a zatim s obzirom na to po kojoj osi se pomakne joystick palimo određenu lampicu. Svaka lampica ima svoju boju.</p> <p>RGB vrijednosti su postavljene na 150 kako bi se malo ublažio intenzitet svijetla.</p>
--	---

## 2. korak

Orion pločica sama pali lampice slučajnim redoslijedom. Pri tome pamti redoslijed kojim pali lampice i svaki puta kada doda novu lampicu u niz ponavlja niz paljenja lampica kako to ide u igri „Simon“.

Lampice su označene brojevima od 1 do 4. Jedan takav niz paljenja lampica može izgledati ovako:

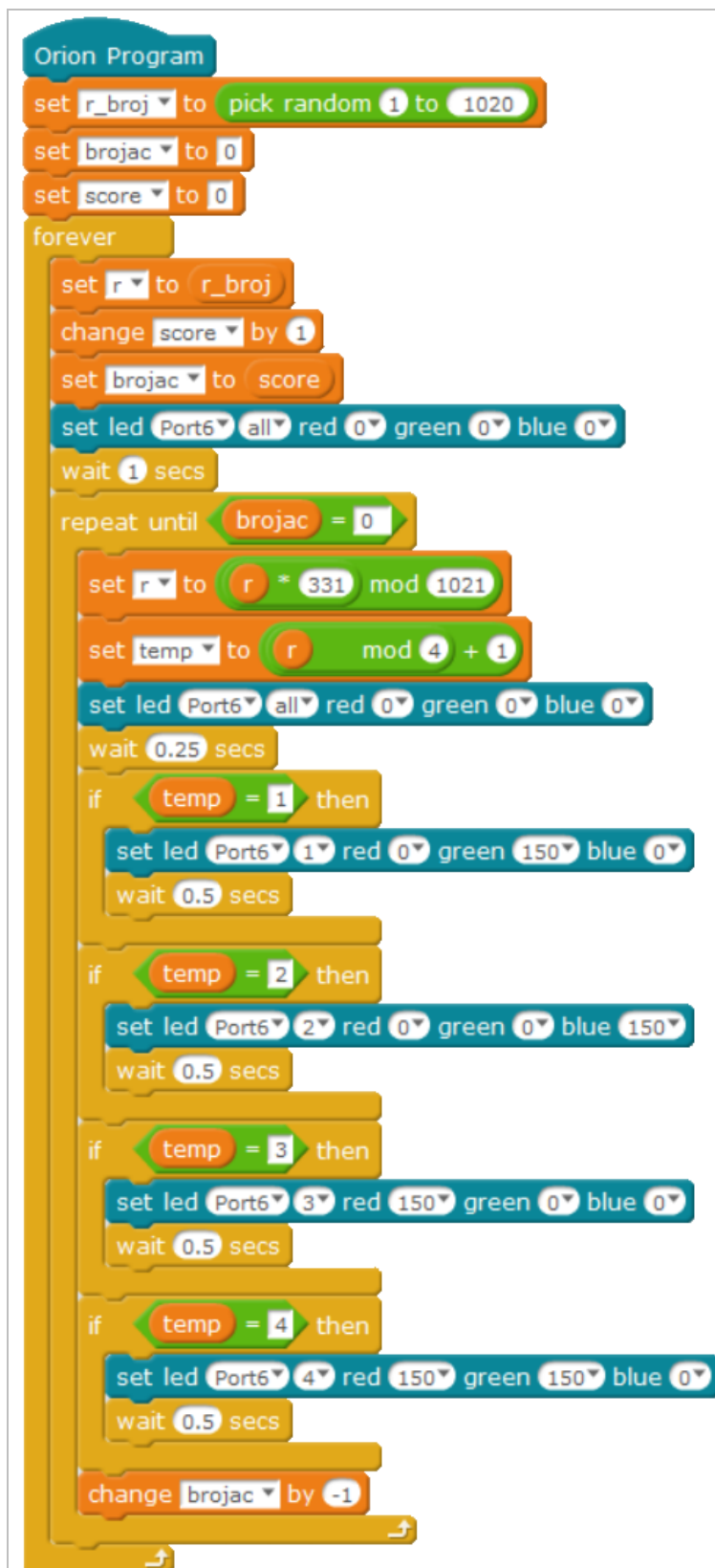
[2] [2, 3] [2, 3, 2] [2, 3, 2, 1] [2, 3, 2, 1, 4] [2, 3, 2, 1, 4, 4]

Budući da mBlock nema podršku za liste ili stringove kod programiranja Orion pločice potrebni su neki alternativni načini za pamćenje redoslijeda kojim se pale lampice.

Jedan od načina je korištenje Linearnog kongruentnog generatora.

([https://en.wikipedia.org/wiki/Linear\\_congruential\\_generator](https://en.wikipedia.org/wiki/Linear_congruential_generator))

Ukratko, na ovaj način se mogu generirati dovoljno dobri pseudo random brojevi koje nije potrebno „pamtiti“ nego se mogu po potrebi izgenerirati u istom redoslijedu.



Prvi slučajan broj odaberemo pomoću bloka dostupnog u mBlocku, a svaki sljedeći računamo rekurzivno.

$$x_{n+1} = 331x \bmod 1021$$

Ovako bi u Pythonu izgledalo generiranje 10 brojeva od 1 do 4 kakvo koristimo i za ovu igru:

```
import random
r = random.randint(1, 1020)
for i in range(10):
    r = (r * 331) % 1021
    print ((r % 4) + 1)
```

Prilikom svakog ponavljanja ponovno generiramo brojeve koristeći početno odabrani broj. Svaki puta generiramo jedan broj više.

Palimo odgovarajuće LED lampice.

### 3. Korak

Spojiti 1. i 2. korak, provjeriti jesu li brojevi uneseni u ispravnom redoslijedu, brojiti uspješno unesene nizove, prikazati ih na 7-segmentnom displayu i zaokružiti igru kao cjelinu.

## Orion Program

forever

set led Port6 all red 60 green 0 blue 60

if joystick Port8 Y-Axis > 400 then

set r\_broj to pick random 1 to 1020

set brojac to 0

set score to -1

set gotovo to 0

set signal to 1

set led Port6 all red 0 green 0 blue 0

wait 2 secs

repeat until gotovo = 1

set 7-segments display Port4 number score

set r to r\_broj

change score by 1

set brojac to score

set led Port6 all red 0 green 0 blue 0

wait 1 secs

repeat until brojac = 0

set r to  $r = 331 \text{ mod } 1021$

set temp to  $r \text{ mod } 4 + 1$

set led Port6 all red 0 green 0 blue 0

wait 0.25 secs

if temp = 1 then

set led Port6 1 red 0 green 150 blue 0

wait 0.5 secs

if temp = 2 then

set led Port6 2 red 0 green 0 blue 150

wait 0.5 secs

if temp = 3 then

set led Port6 3 red 150 green 0 blue 0

wait 0.5 secs

if temp = 4 then

set led Port6 4 red 150 green 150 blue 0

wait 0.5 secs

change brojac by -1

Na početku sve lampice svijetle u ružičastoj boji. Čeka se na pomak joysticka prema gore kako bi igra započela

Dodali smo prikaz rezultata na 7-segmentni zaslon.

```

set r to r_broj
set brojac to score
repeat until gotovo = 1 or brojac = 0
  set led Port6 all red 0 green 0 blue 0
  if signal = 1 then
    set r to r * 331 mod 1021
    set temp to r mod 4 + 1
    set signal to 0
  if joystick Port8 X-Axis > 400 then
    set led Port6 1 red 0 green 150 blue 0
    wait 0.5 secs
    if not temp = 1 then
      set gotovo to 1
    else
      change brojac by -1
      set signal to 1
  if joystick Port8 X-Axis < -400 then
    set led Port6 3 red 150 green 0 blue 0
    wait 0.5 secs
    if not temp = 3 then
      set gotovo to 1
    else
      change brojac by -1
      set signal to 1
  if joystick Port8 Y-Axis > 400 then
    set led Port6 4 red 150 green 150 blue 0
    wait 0.5 secs
    if not temp = 4 then
      set gotovo to 1
    else
      change brojac by -1
      set signal to 1
  if joystick Port8 Y-Axis < -400 then
    set led Port6 2 red 0 green 0 blue 150
    wait 0.5 secs
    if not temp = 2 then
      set gotovo to 1
    else
      change brojac by -1
      set signal to 1

```

Kod blokova koji određuju paljenje lampica pomoću joysticka dodali smo i provjeru pale li se one u ispravnom redosljedu. Pri tome generiramo potrebni niz bojeva rekursivno kako je ranije opisano.

Ako se odabere pogrešna lampica igra završava.

Pomoću varijable *signal* ostvarujemo čekanje na odabir lampice pomoću joysticka prije nego generiramo sljedeći broj.

Varijabla *brojac* pamti koliko je još brojeva potrebno izgenerirati u trenutnoj rundi igre.