

Programiranje  
i robotika  
u radu s djecom

Ivan Bosnić, MIPS d.o.o.



JavaCro



MIPS®



# Sadržaj

---

---

Osobno iskustvo

---

Vrtić/predškola

---

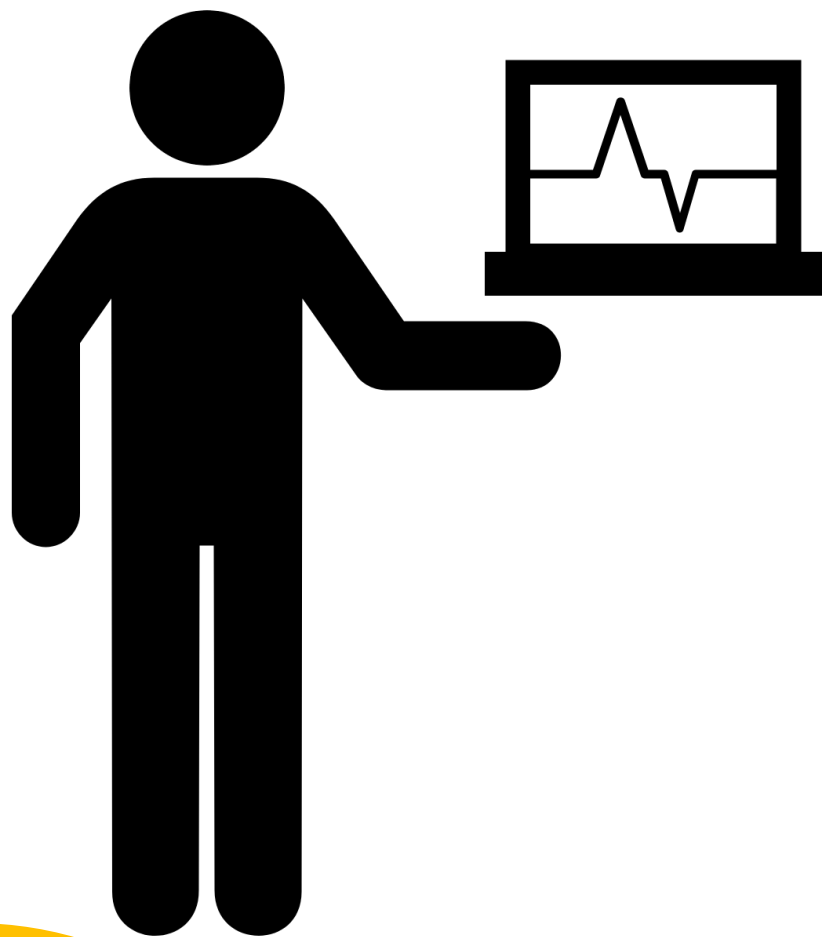
Razredna nastava

---

Predmetna nastava

---

Natjecanja

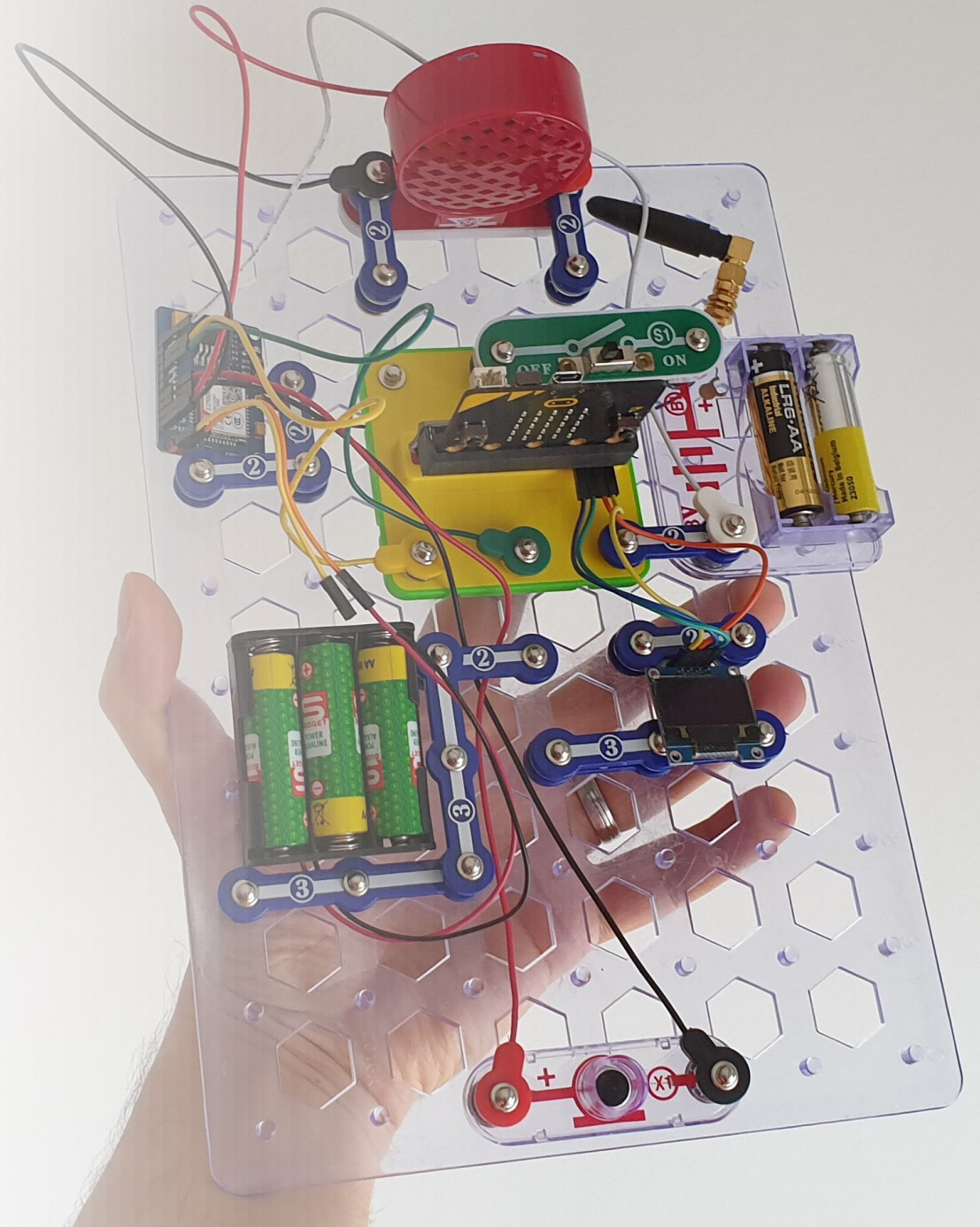


## Osobno iskustvo

- od 2017. u neformalnom obrazovanju
- od igre sa sinom do edukacija
- puno naučenog
- brojna poznanstva
- mnogo osmijeha :)

# Savjeti

- dječja individualnost
- učenje igrom
- predstavljanje umjesto nametanja
- višeslojne aktivnosti
- posjećivanje STEM događanja
- „manje je više”
- „sharing is caring”



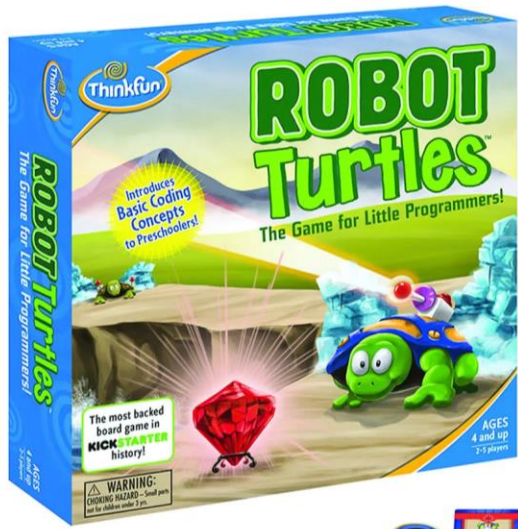


## Vrtić/predškola

- izazovnije zbog dobi djece - malo vrtića i ustanova nudi aktivnosti
- mogućnosti:
  - učenje kroz igru
  - uključivanje djetetovih prijatelja
  - opremanje grupe u vrtiću
  - organiziranje večeri programiranja



# Robot Turtles

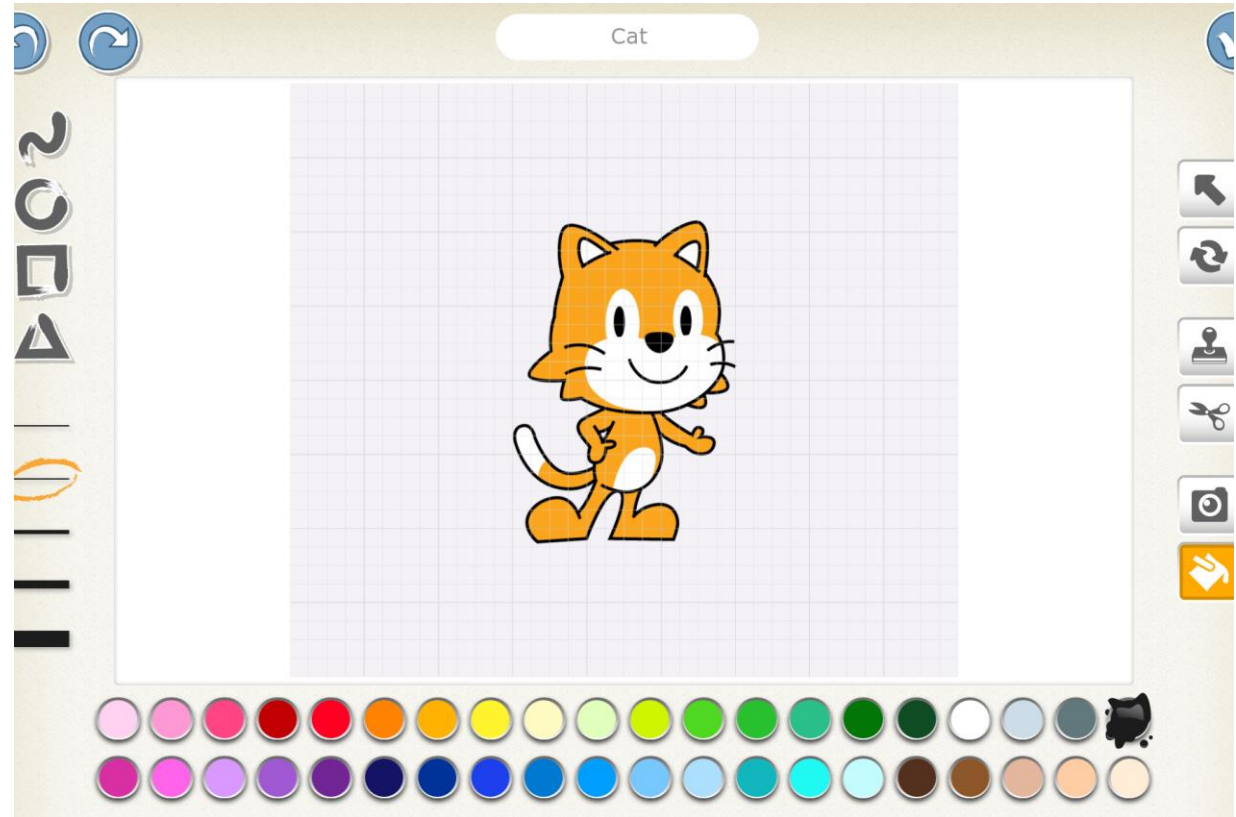
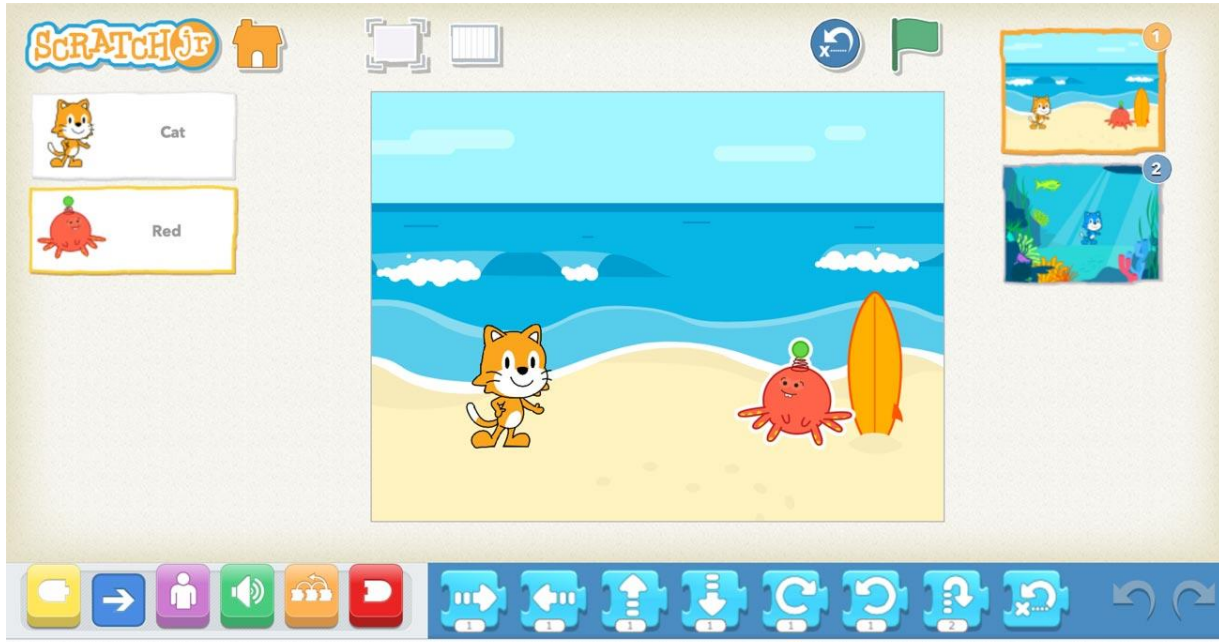








# Scratch Jr





# Osmo Coding





# Bee-Bot / Botley 2.0 the Coding Robot / Code & Go Robot Mouse Activity Set / Matatalab / Pro-Bot





# Scottie Go!

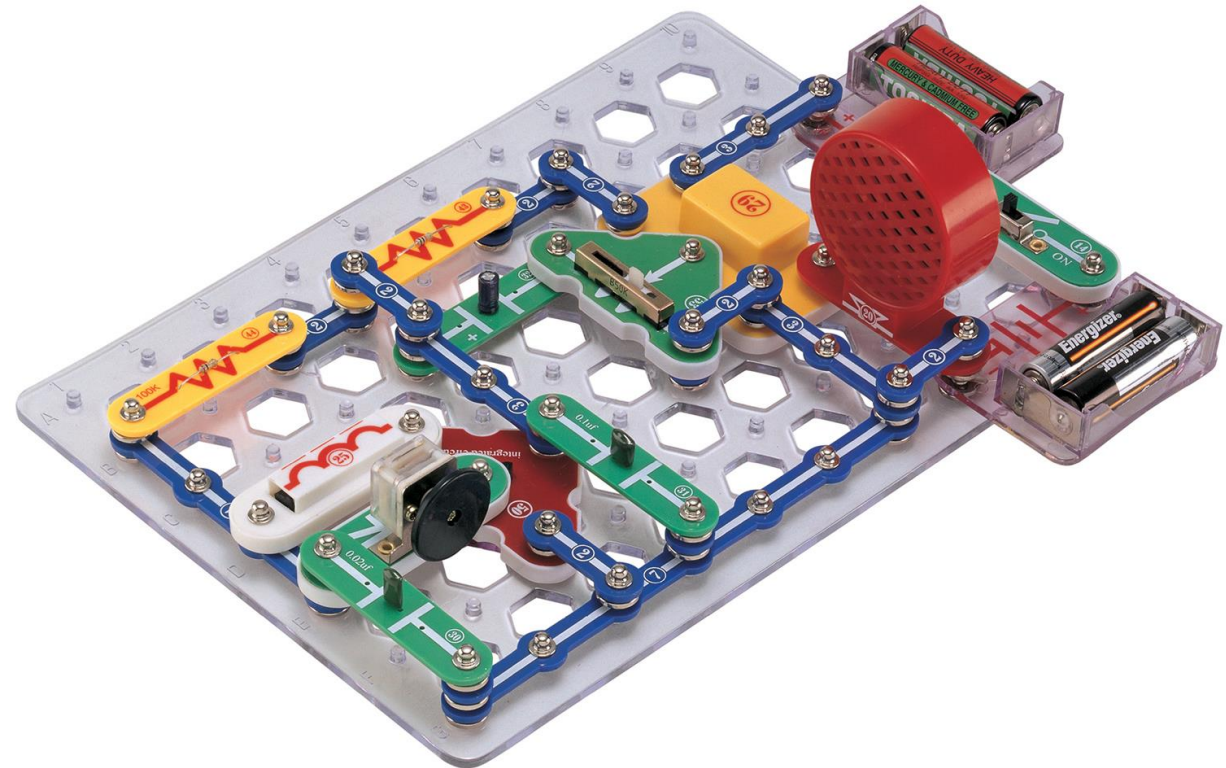




# Run Marco!



# ⚡ Snap Circuits



# ⚡ Makey Makey



# Spintronics







# Razredna nastava

- Scratch u školama
- razne aktivnosti i natjecanja
- mogućnosti:
  - lokalizacija alata i materijala
  - mentoriranje, izravno ili kroz rad neke ustanove



# Ozobot





# Potato Pirates





# Scratch

The screenshot shows the Scratch code editor interface. At the top, there's a purple header with the Scratch logo, settings, and a 'Join Scratch' button. Below the header, there are tabs for 'Code', 'Costumes', and 'Sounds'. The left sidebar contains a vertical menu with categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The 'Sound' category is selected, showing a list of sound-related blocks: 'play sound Chirp until done', 'start sound Chirp', 'stop all sounds', 'change pitch effect by 10', 'set pitch effect to 100', 'clear sound effects', 'change volume by -10', and 'set volume to 100 %'. The main workspace shows a green dinosaur sprite on a grid. A script block is attached to the 'when green flag clicked' event, containing a 'play sound Chirp until done' block. At the bottom, there's a 'Sprite' panel for 'Sprite1' with coordinates (0, 0), size 100, and direction 90.

The screenshot shows the 'Choose an Extension' screen in Scratch. It features a purple header with a 'Back' button and the text 'Choose an Extension'. Below the header, there are eight extension cards arranged in a 2x4 grid. Each card has a colorful background, an icon, a title, a description, and a 'Requires' section.

- Music**: Play instruments and drums. Requires: Internet connection.
- Pen**: Draw with your sprites. Requires: Internet connection.
- Video Sensing**: Sense motion with the camera. Requires: Internet connection.
- Text to Speech**: Make your projects talk. Requires: Internet connection. Collaboration with Amazon Web Services.
- Translate**: Translate text into many languages. Requires: Internet connection. Collaboration with Google.
- Makey Makey**: Make anything into a key. Requires: Internet connection. Collaboration with JoyLabz.
- micro:bit**: Connect your projects with the world. Requires: Internet connection. Collaboration with micro:bit.
- LEGO MINDSTORMS EV3**: Build interactive robots and more. Requires: Internet connection. Collaboration with LEGO.

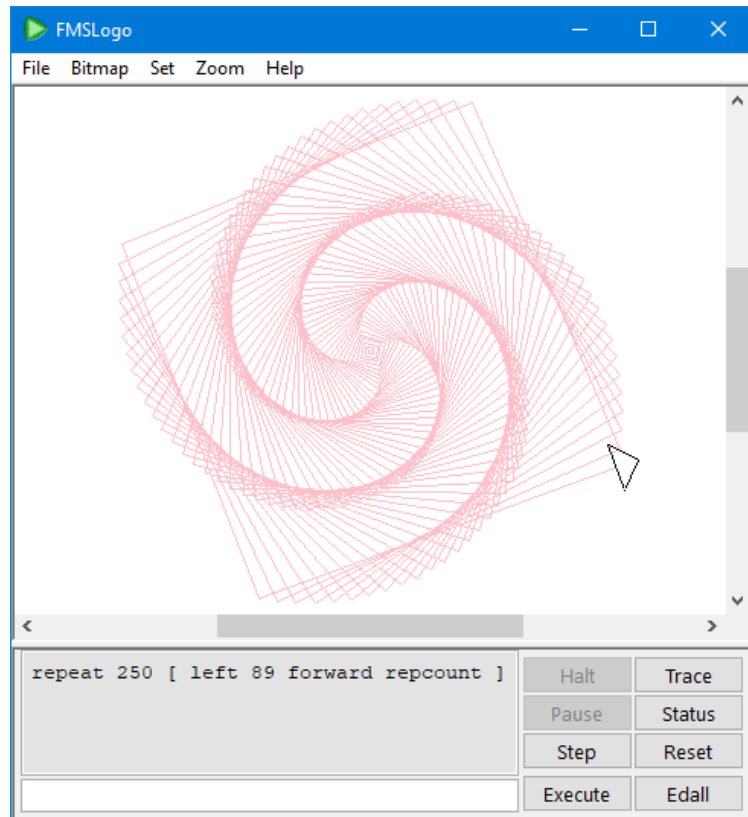


# LEGO WeDo 2.0





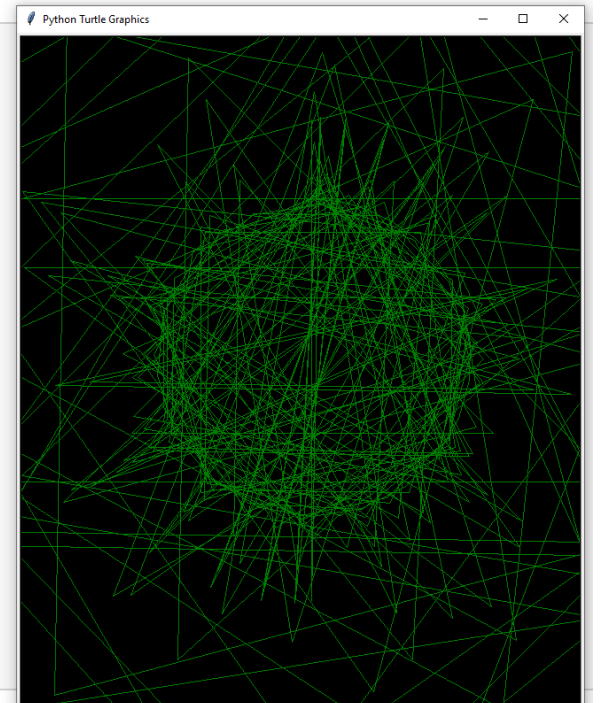
# Logo



## Virus in Python (Turtle Library)

```
import turtle
a=0
b=0
#clcoding.com
turtle.bgcolor("black")
turtle.speed(0)
turtle.pencolor("green")
turtle.penup()
turtle.goto(0,200)
turtle.pendown()

while True:
    turtle.forward(a)
    turtle.right(b)
    a+=3
    b+=1
```



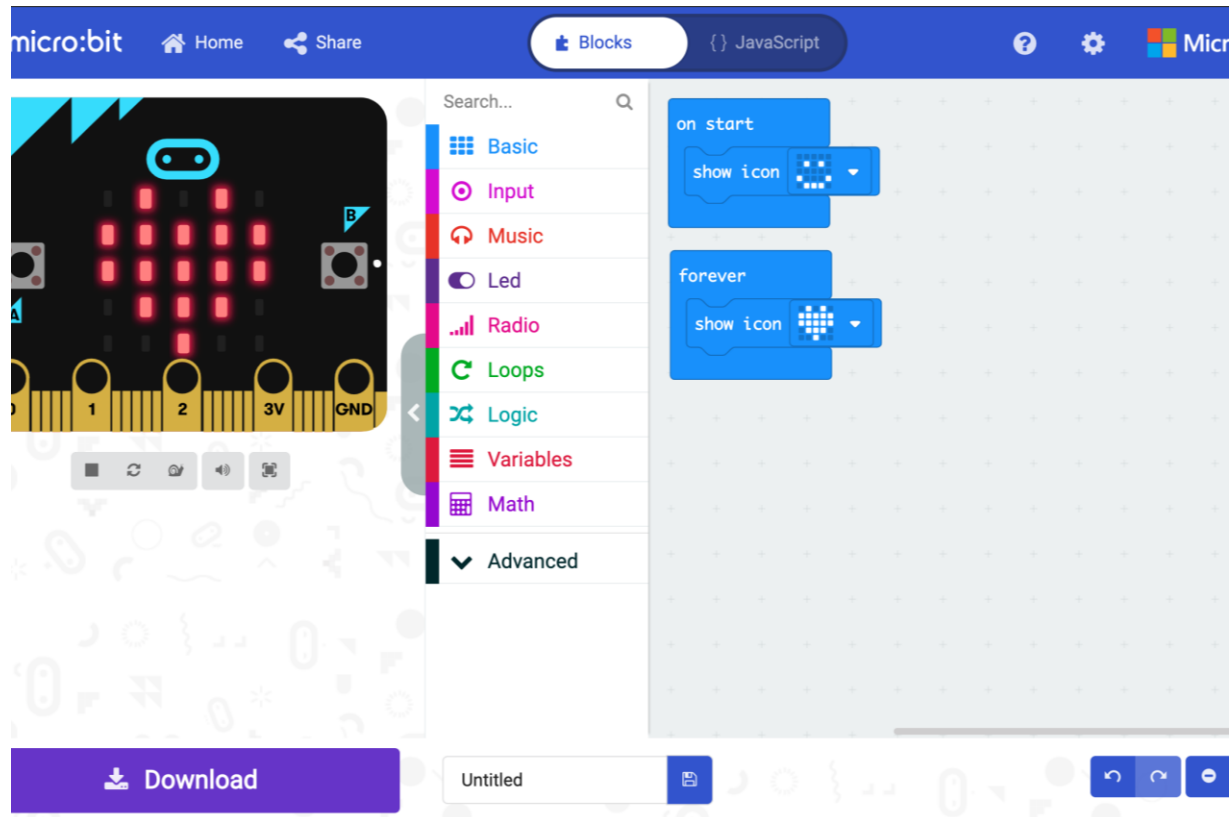


# Turing Tumble





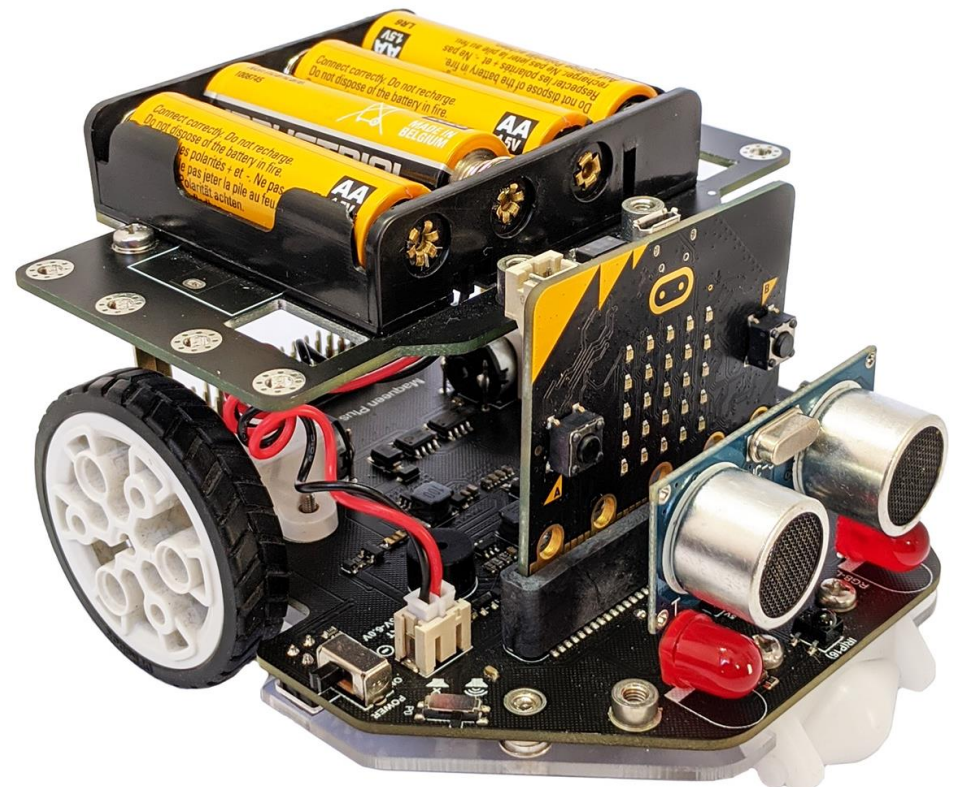
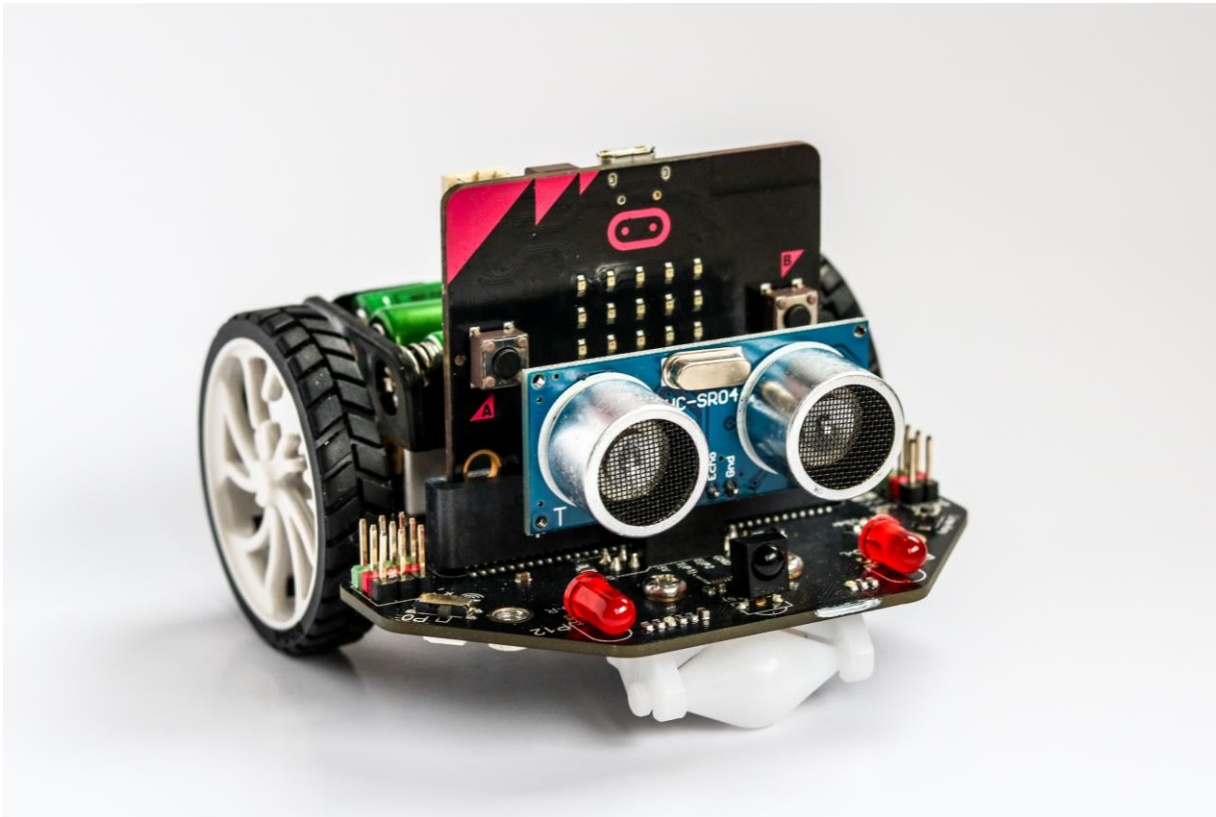
# MakeCode for micro:bit







# Maqueen





# Microblocks

The screenshot shows the Microblocks IDE interface. On the left, there is a sidebar with categories: Output, Input, Pins, Control, Operators, Variables, Data, My Blocks, Libraries, Basic Sensors, and TFT. The main workspace contains a script starting with "when started", followed by "clear display", and a "forever" loop. Inside the loop, there are blocks for "set coordX to" and "set coordY to" using mathematical formulas:  $\text{tilt } x + 50 \times \text{TFT width} / 100$  and  $\text{tilt } y + 50 \times \text{TFT height} / 100$ . This is followed by "draw circle on TFT at x coordX y coordY radius 5 color" (with a white circle icon) and "wait 20 millisecs", then another "draw circle on TFT at x coordX y coordY radius 5 color" (with a black circle icon). Below the main script, there is another "forever" loop containing a "graph tilt x tilt y" block. In the bottom-left corner, a "Data Graph" window is open, showing a line graph with two data series (green and red) plotted against a y-axis ranging from -100 to 100. The graph shows oscillating signals.

The screenshot shows the Microblocks IDE interface for a project named "Heartbeat". On the left, there is a sidebar with categories: a command block and a reporter block. The main workspace contains a script starting with "when button A pressed", followed by a "repeat until not button A" loop. Inside the loop, there are blocks for "display" (showing a grid of red and blue squares), "wait 250 millisecs", "display" (showing a grid of red and blue squares), "wait 250 millisecs", "clear display", and "wait 300 millisecs". On the right, there is a visual representation of a yellow board labeled "Boardie" with a "RESET" button and a power icon. The board displays a 5x5 grid of green squares on a black background, with a pattern of green squares forming a cross-like shape. Below the board, there are three buttons labeled "A", "A+B", and "B".



# Minecraft Education

```
on chat command "wander"
  agent teleport to player
  agent place on move true
  agent destroy obstacles true
  repeat 100 times
    do
      set item to pick random 0 to 10
      if item <= 1 then
        agent turn left
      else if item <= 3 then
        agent turn right
      else
        agent move forward by 1
```



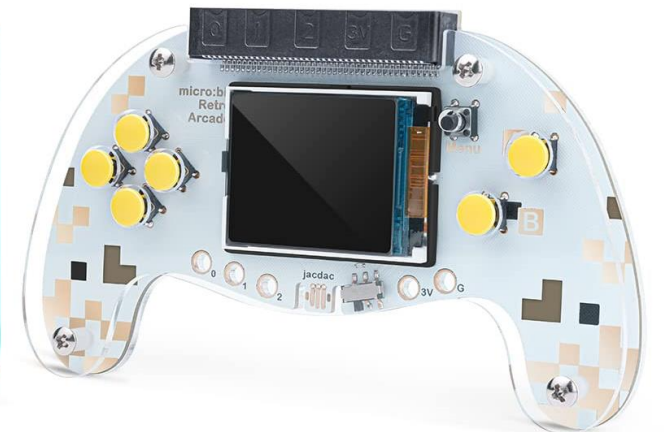
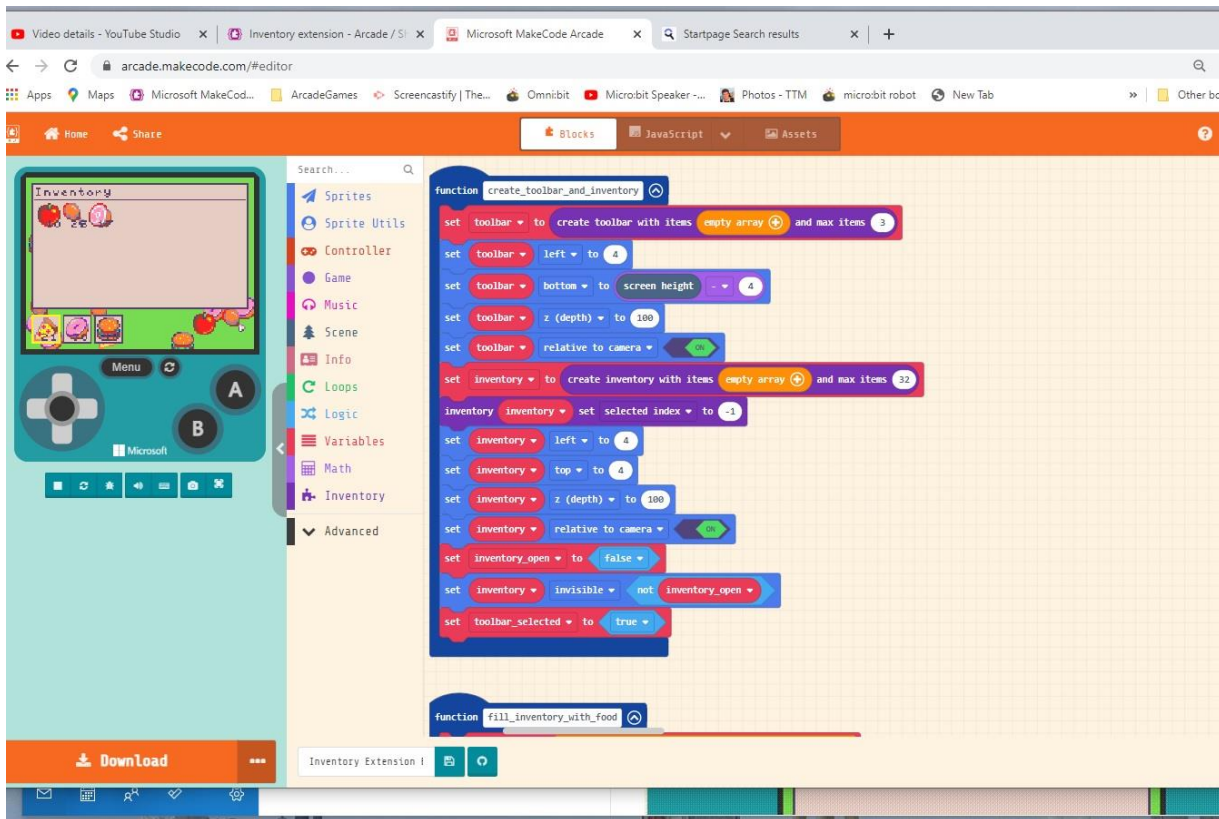


# Predmetna nastava

- većinom Python i HTML
- dronovi i umjetna inteligencija
- sve je više aktivnosti i natjecanja
- mogućnosti:
  - pomoć učiteljima
  - izrada novih alata i sadržaja ili proširenje postojećih

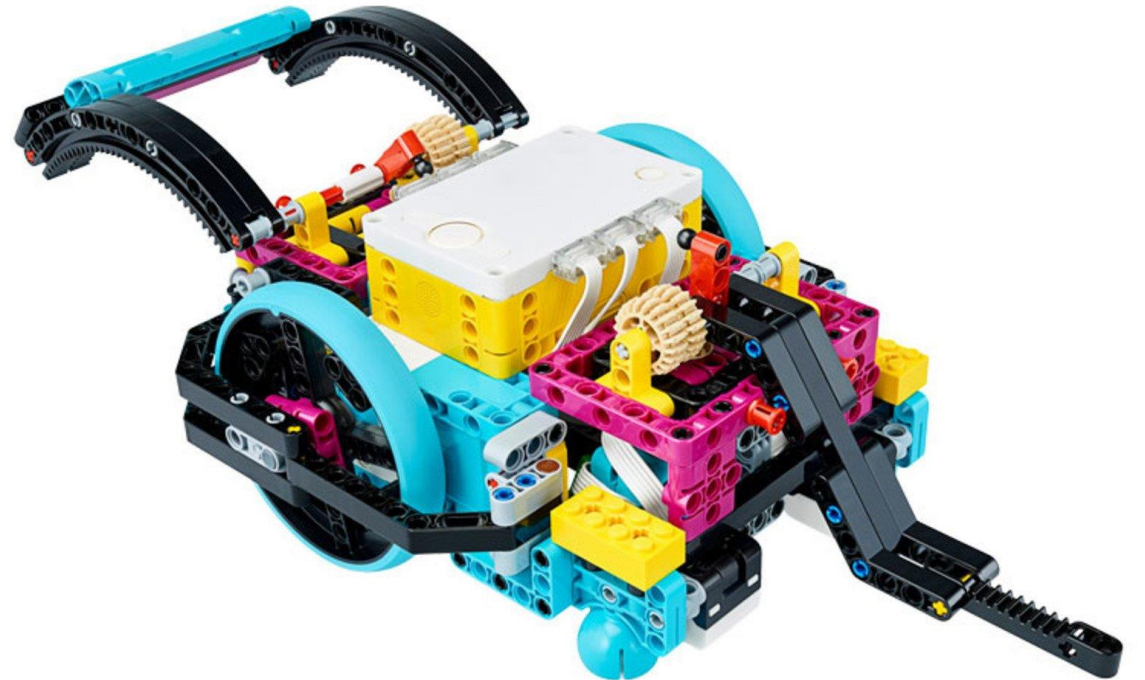


# MakeCode Arcade





# LEGO SPIKE Prime





# MIT App Inventor

This screenshot shows the MIT App Inventor interface in the Designer view. The central canvas displays a mobile app interface for a maze game. At the top, it shows the name "2C 07 Lai Pak Chuen Patrick" and the time "9:48". The main area is a maze with a yellow player icon at the entrance. Below the maze are three buttons: "Random maze?", "Time: 00 : 00 : 00", and "Start Pause Reset". The interface is surrounded by several panels: "Palette" on the left with a search bar and categories like "User Interface", "Layout", "Media", "Drawing and Animation", and "Maps"; "Components" on the right showing a tree view of the app's structure; and "Properties" on the far right showing settings for the selected "Screen1" component, such as "AboutScreen", "AccentColor", "AlignHorizontal", "App Name", and "BackgroundImage".

This screenshot shows the same MIT App Inventor interface but in the Blocks view. The central canvas is a workspace for visual programming, where the app's logic is defined using colored blocks. The "Blocks" panel on the left lists various built-in components like "Control", "Logic", "Math", "Text", "Lists", "Dictionaries", "Colors", "Variables", "Procedures", and "Screen1". The workspace contains several event-driven blocks: "when Random\_maze? Click" triggers "set maze2.png to maze.png" and "make a list"; "when Start\_timer Click" triggers "set Time\_minutes to 00" and "set Time\_seconds to 00"; "when End\_game Click" triggers "set Time\_minutes to 00" and "set Time\_seconds to 00"; and "when Screen1.Timer" triggers a complex sequence of blocks including "set Time\_minutes to 00", "set Time\_seconds to 00", and "set Time\_hours to 00". The "Properties" panel on the right is hidden, and the "Media" panel at the bottom shows the assets used in the app.



# Python

The screenshot shows the Mu Python IDE interface. At the top, there is a menu bar with icons for Mode, New, Load, Save, Serial, Plotter, Zoom in, Zoom out, Theme, Check, Help, and Quit. Below the menu bar, a dialog box titled "Select Mode" is open, asking the user to select a mode. The dialog lists four options:

- Adafruit CircuitPython: Use CircuitPython on Adafruit's line of boards.
- BBC micro:bit: Write MicroPython for the BBC micro:bit.
- Pygame Zero: Make games with Pygame Zero.
- Python 3: Create code using standard Python 3.

At the bottom of the dialog, there are "Cancel" and "OK" buttons. The main code editor area is visible in the background, showing a single line of code: "# Write your code".

The screenshot shows the app.edublocks.org website. The browser address bar displays "app.edublocks.org". The page title is "Turtle Spiral". The interface is split into two views: "Blocks" and "Code".

The "Blocks" view shows a sequence of code blocks for a Turtle Spiral program:

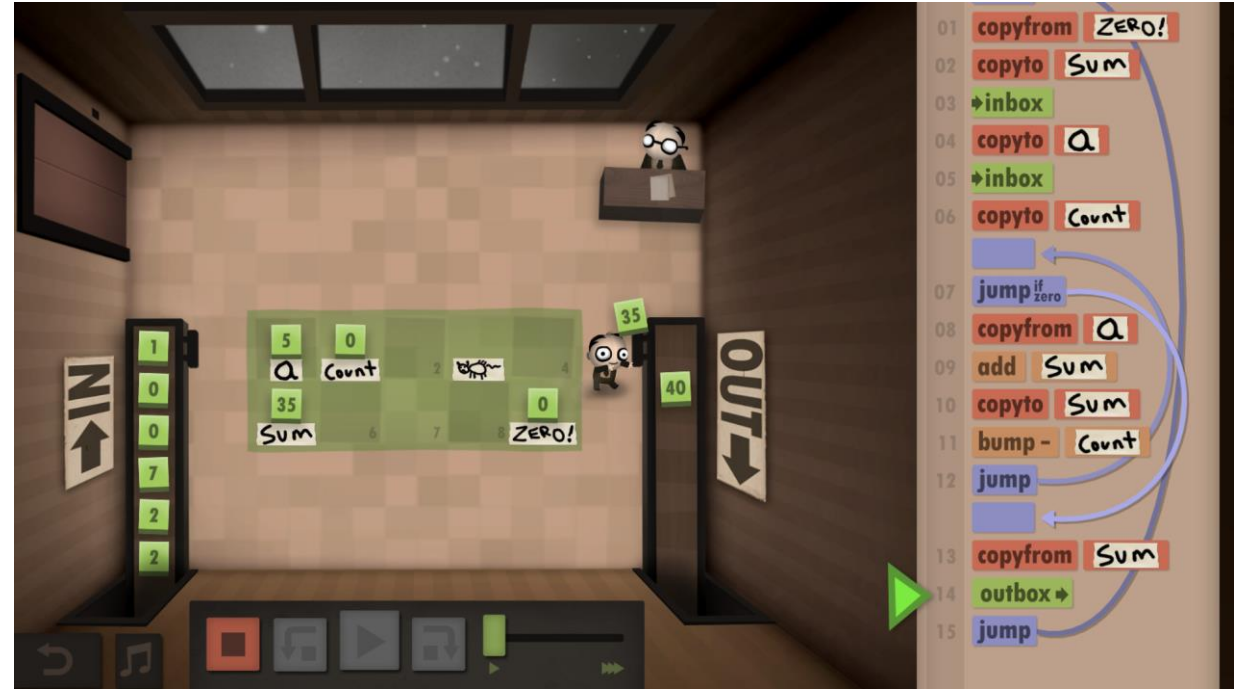
- # Start Code Here
- from turtle import \*
- turtle = Turtle()
- screen = Screen()
- screen .setup( 1000,1000 )
- screen .bgcolor( "black" )
- turtle .speed( 0 )
- colours = [ "red", "purple", "blue", "green", "orange", "yellow" ]
- for ( i in range( 360 ) ):
  - turtle .color( colours [ i % 6 ] )
  - turtle .width( i / 100 + 1 )
  - turtle .forward ( i )
  - turtle .left ( 59 )

```
1 # Start Code Here
2 from turtle import *
3 turtle = Turtle()
4 screen = Screen()
5 screen.setup(1000,1000)
6 screen.bgcolor("black")
7 turtle.speed(0)
8 colours = ["red", "purple", "blue", "green", "orange", "yellow"]
9 for i in range(360):
10 turtle.color(colours[i % 6])
11 turtle.width(i / 100 + 1)
12 turtle.forward(i)
13 turtle.left(59)
14
```





# Human Resource Machine





# Tinkercad

The screenshot shows the Tinkercad workspace with a 3D model of a DNA double helix. The left sidebar contains a 'Shapes' panel with various geometric primitives like Box, Cylinder, Sphere, Cone, and Text. The central workspace shows a 'Primitives Meter' at the top with a speed slider and a 'Step ID' button. The right side of the workspace displays a block-based code editor with the following sequence of blocks:

- Create New Object sugar
- Add Sides 5 edge
- Scale: X: 0.1 Y: 0.1 Z: 0.05
- Rotate around Axis y by 90 Degrees
- Create New Object object0
- Repeat 32 Times
- Add Copy of Object sugar
- Rotate around Axis z by rotation Degree
- Move: X: radius \* Cos angle
- Change angle by spacing
- Change height by 3
- Change rotation by 8
- Create Group
- Create New Object object1
- Add Copy of Object object0
- Rotate around Axis z by 180 Degrees
- Create Group
- Delete Object sugar

The screenshot shows a Tinkercad circuit simulation. The circuit includes a breadboard with an Arduino Uno, a motor, and a battery pack of three AA 1.5V batteries. The code editor on the right contains the following blocks:

- on start
- clear screen
- show leds
- wait 2 secs
- clear screen
- show number 0
- show string Hello!
- plot bar graph of 0 up to 0
- wait 1 secs
- on button A pressed
- digital write pin P2
- wait 5 secs



# Processing

The screenshot shows the Processing IDE with a code editor on the left and a preview window on the right. The code editor displays the following code:

```
1 /**
2  * A Processing implementation of Game of Life
3  * By Joan Soler-Adillon
4  *
5  * Press SPACE BAR to pause and change the cell's values with the mouse
6  * On pause, click to activate/deactivate cells
7  * Press R to randomly reset the cells' grid
8  * Press C to clear the cells' grid
9  *
10 * The original Game of Life
11 */
12
13 // Size of cells
14 int cellSize = 5;
15
16 // How likely for a cell to be alive
17 float probabilityOfAliveAtStart = 0.5;
18
19 // Variables for timer
```

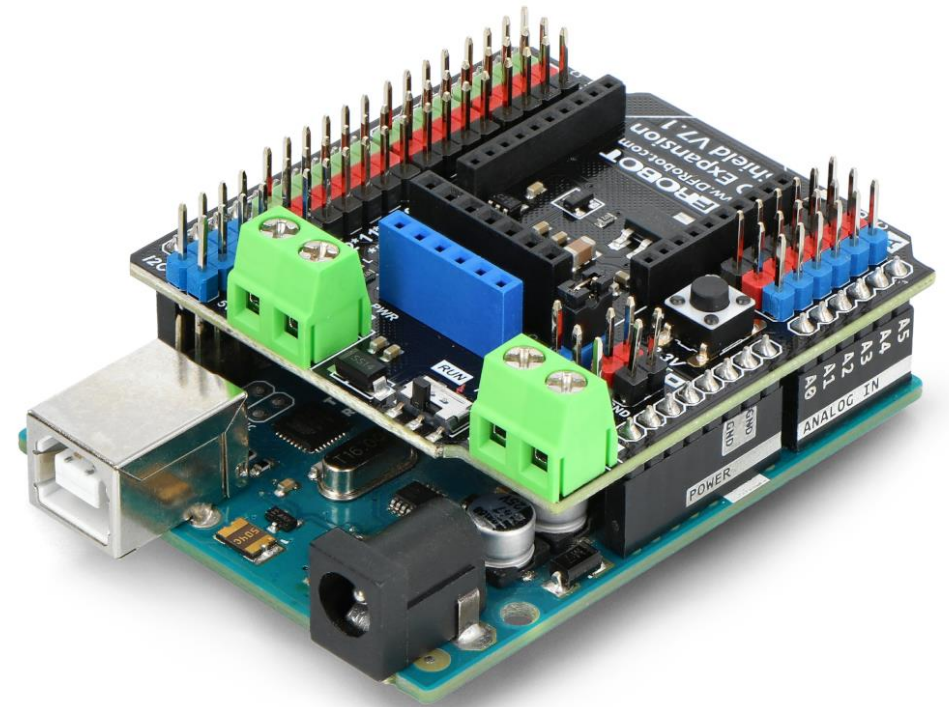
The preview window, titled "GameOfLife", shows a 2D grid of cells. Some cells are green, representing a pattern of life, while others are black. The background of the preview window is a dark blue space with white stars.

The screenshot shows the p5.js Web Editor interface. The code editor on the left contains the following code:

```
1 var teapot1;
2 var teapot2;
3 var cup;
4
5 var teapotLayers = [];
6
7 var lastPosition;
8 var stillCount;
9
10 var colorWidth = 40;
11 var colorOffset = 0;
12
13 function setup() {
14   createCanvas(windowWidth, windowHeight, WEBGL);
15   lastPosition = new p5.Vector(mouseX, mouseY);
16   teapot1 = loadModel('teapot.obj');
17   teapot2 = loadModel('crackery_pot.obj');
18   cup = loadModel('koffie.obj');
19
20   directionalLight(125, 125, 125, -1, 1, 0.5);
21   ambientLight(125);
22   colorMode(HSB, 255);
23 }
24
25 function draw() {
26   if (dist(lastPosition.x, lastPosition.y, mouseX, mouseY) > 0) {
27     stillCount = 0;
28     colorOffset = floor(random(255));
29   } else {
30     if (stillCount >= 10) {
31       if ((stillCount - 10) % 60 == 0) {
32         var newWidth = width / height * 900 * 0.5;
33         var newX = map(mouseX, 0, width, -newWidth, newWidth);
34         var newY = map(mouseY, 0, height, -450, 450);
35
36         var layer = [];
37         layer.center = new p5.Vector(newX, newY);
38         layer.startFrame = frameCount;
39         layer.axis = (random(1) < 0.5) ? 'x' : 'y';
40         layer.rotation = random(0.1);
41         layer.orbit = random(0.05) * 0.025;
42         layer.number = floor(random(5, 30));
43       }
44     }
45   }
46 }
```

The preview window on the right shows a 3D scene with a white background. It features several teapots and a cup, each with a different color (red, green, yellow, purple). The objects are arranged in a circular pattern, and some are slightly offset from the center. The scene is lit with a directional light and an ambient light.

# ⚡ Gravity





# Plasma



**ROBOTIC ARM :: CONTROL**

MINIMAP LABEL VARIABLES  
CONSOLE INTERFACE

Search...

**COMPONENTS**

- BUTTON
- CLAW
- CONTROLLER
- LEVER
- SERVO
- SINGLE AXIS JOINT
- SINGLE AXIS JOINT 02
- SLIDER

**ROTATION**  
Slider rotates the Arm 90° from left to right

**SLIDER**  
Output Position: 38.00%  
Position: 38.00%  
Mouse Mapping: Horizontal  
LED Strip Color: [Blue]

**PERCENTAGE MAPPER**  
Map: [Slider] → Number: [Servo]  
Percentage: 38.00%  
From: 45.00  
To: -45.00  
Use Negative Limit: [ ]  
Decoration: Generic  
Ease: Linear

**SERVO**  
<NONE> Rotation Reached  
Current Rotation  
Target Rotation: 10.80  
Max Speed: 50.00%  
Torque: 50.00%

**JOINTS**  
The Lever controls both the joints...  
Could be automated with distance sensor though (?)

**LEVER**  
Output Position: 40.50%  
Position: 40.50%  
LED Strip Color: [Blue]

**PERCENTAGE MAPPER**  
Map: [Lever] → Number: [Joint 01] & [Joint 02]  
Percentage: 40.50%  
From: 0.00  
To: 90.00  
Use Negative Limit: [ ]  
Decoration: Generic  
Ease: Linear

**SINGLE AXIS JOINT**  
<NONE> Target Reached  
Target Rotation: 63.23  
Max Speed: 50.00%  
Torque: 50.00%

**SINGLE AXIS JOINT 02**  
<NONE> Target Reached  
Target Rotation: 63.23  
Max Speed: 50.00%  
Torque: 50.00%



# Arduino

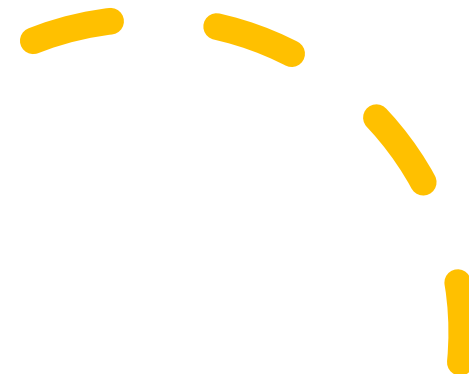
```
Blink | Arduino 1.8.5  
Blink §  
This example code is in the public domain.  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

32 Arduino/Genuino Uno on COM1





Natjecanja





# Natjecanje iz informatike (5.+)



- <https://informatika.azoo.hr/>
- školsko natjecanje
- algoritmi
  - Logo
  - Python/C/C++
- Osnove informatike
  - Digitalne kompetencije (5. i 6.)
  - Osnove informatike (7. i 8.)
- Razvoj softvera
  - od ideje do softvera





# HONI (3.+)



Hrvatsko otvoreno natjecanje u informatici

- Hrvatsko otvoreno natjecanje u informatici
- <https://hsin.hr/honi/>
- online natjecanje
- 5 do 7 kola tijekom školske godine
- Python, Pascal, C/C++ i Java
- najbolji natjecatelji idu na Juniorsku hrvatsku informatičku olimpijadu



# HLL (1.+)



HRVATSKA LOGO LIGA

- Hrvatska logo liga
- <https://logoliga.hsin.hr/>
- online natjecanje
- 5 do 6 kola kroz školsku godinu
- FMSLogo
- najbolji natjecatelji idu na Hrvatsku logo olimpijadu



# Dabar (1.+)



- <https://ucitelji.hr/dabar/>
- međunarodni školski izazov iz informatike i računalnog razmišljanja
- jednom godišnje online se rješava 12 zadataka
  - MikroDabar (1. i 2.)
  - MiliDabar (3. i 4.)
  - KiloDabar (5. i 6.)
  - MegaDabar (7. i 8.)

 FLL (2.+)

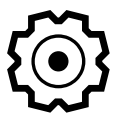


- FIRST LEGO League
- <https://fllcroatia.org/>
- ekipa s 3 do 10 članova
- dobije se zadatak i staza, slaže se i programira LEGO Technic robot za rješavanje
- Challenge (4.-8.), Explore (2.-4.) i Discover (4-6 godina, zasad nema u Hrvatskoj)
- kotizacija

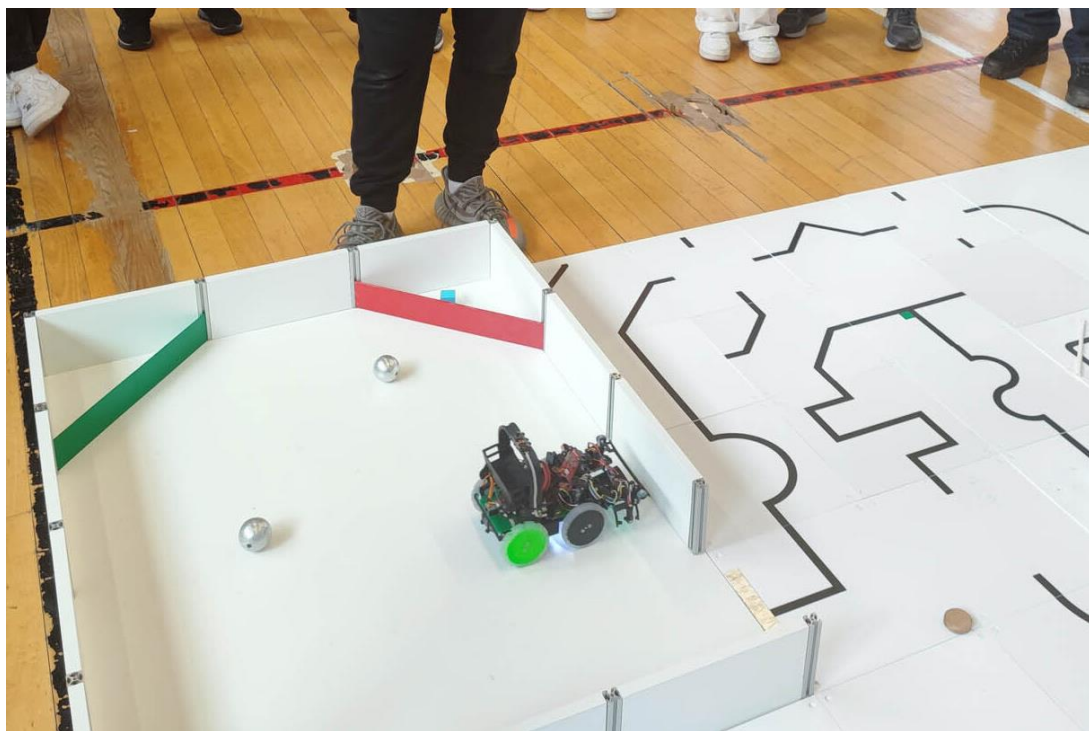
 WRO (2.+)



- World Robot Olympiad
- <https://wro.hr/>
- ekipa s 2 do 3 člana
- napraviti i programirati LEGO robota koji savladava zadatke na podlozi (Robomission i Future Inovators) ili se natječe protiv drugih robota (Robosports)
- kotizacija



# RoboCup Junior (1.+)



- <https://robocupcroatia.com/>
- ekipa s 2 do 4 člana
- izrada robota koji spašava žrtvu (Rescue), izvodi koreografiju (OnStage) ili igra nogomet (Soccer)
- kotizacija



# Croatian Makers liga (1.+)



- <https://croatianmakers.hr/en/croatian-makers-league/>
- ekipa s 2 do 4 člana
- dijelom online, a dijelom uživo
- kroz niz kola rješavaju se razni zadaci na stazi s robotom Maqueen
- najbolje ekipe idu u superfinale



# Generacija NOW (1.+)



- <https://croatianmakers.hr/hr/generacija-now/>
- projekt
- ekipe učenika dobiju opremu, poduku i materijale te rješavaju zadane i vlastite izazove
- najbolji radovi budu izloženi i nagrađeni na završnome događanju



# Škola budućnosti (1.+)



- <https://skolabuducnosti.stemi.education/>
- projekt
- uključenim školama donira se oprema i materijali, imaju pristup edukacijama i predavanjima te mogu sudjelovati na završnoj konferenciji

Zahvaljujem na pozornosti.