



Robot Academy mBot2 Robot Welcome Guide

Table of Contents

1. Tour of MBot2 Robot.....	2
2. Jousting Program	3
3. BattleBot	4
4. AI with mBot2	6
A. Voice Recognition	6
B. Translation	6
5. Video Game Design	7
6. Troubleshooting Guide	10
A. Installation.....	10
B. Robot Won't Turn On.....	10
C. Robot Not Responding to Commands	10
D. Motors Not Working	10
E. Sensors Not Responding	10
F. Additional Tips.....	10
7. Next Steps	11



1. Tour of MBot2 Robot

Welcome to the exciting world of robotics with the Robot Academy mBot2! We are thrilled that your child has joined our robot camp and can explore the fascinating realm of robotics and programming. This guide aims to provide you with an introduction and comprehensive tour of the mBot2.

At the heart of the mBot2 is its sturdy frame, which provides a solid foundation for all its components. The functionality of the robot is powered by the microcontroller, which is practically the brain of the robot. The mBot2 has the CyberPi microcontroller, a highly versatile and powerful component that allows for a wide range of programming and interactive experiences. It is a highly programmable unit, supporting block-based programming for beginners and more advanced languages like Python for advanced users.

The mBot2 is equipped with a range of connectors that allow various components to be attached and powered. These connectors enable the integration of additional modules and sensors, allowing for endless customization and experimentation.

Motors are another crucial component of the mBot2, providing the necessary movement for the robot. The mBot2 includes two powerful encoder motors that can be precisely controlled by their rotation, speed, and position. The sensors play a vital role in the mBot2's ability to interact with its environment. The robot comes equipped with built-in-sensors, such as ultrasonic sensors for obstacle detection, line-following, and light sensors that detect changes in light intensity. These sensors allow the robot to gather information about its surroundings and respond accordingly, enhancing functionality and enabling autonomous performance.

In addition to USB connectivity, the MBot2 Robot has built-in Bluetooth and WiFi capabilities, useful for coding on tablets without USB ports. A reset button can clear the robot's memory if it's still running old code after uploading a new program. The CyberPi also features a power on-off switch and a push button, the latter of which starts some codes.



2. Jousting Program

The jousting program is a simple program that introduces key programming concepts such as events, controls, sensing and the use of mBot2 chassis type blocks, all while providing a fun and interactive experience. The jousting program involves creating a robotic jousting game where mBot2 robots compete in a friendly duel. Each robot is programmed to move, sense its environment, and react to various conditions, stimulating a jousting match.

Events are actions that trigger specific responses from the robot. In the jousting program, we use the block **when CyberPi starts up**, which begins the program when the robot is turned on. Following that, we use a control block. Control structures in programming allow for decision-making and repeated actions. The jousting program uses the control block **wait until** that pauses the programming from carrying out until a circumstance is met. Sensing is the next type of block which elicits the various sensors from that enable the robot to interact with its environment. Our program uses the sensing block, **button B pressed?**, which senses which the B (triangle) is pressed.

Once the button is pressed, commencing the jousting battle, the robots will move forward 50 cm, using the mBot2 chassis block, **moves forward 50 cm until done**. The mBot2 chassis type blocks are specialized programming blocks that control the robot's hardware components. These blocks allow your child to manage the motors to program the robot to move forward, backward, and turn in customizable configurations.



Figure 1 Jousting Program

When the code is done, to upload first turn on the MBot2 Robot and then click upload. After the computer says the code is uploaded, turn off the MBot2 Robot and unplug the USB cable.



3. BattleBot

Battlebots is when two or more robots fight to be the last one standing. We will be coding for a version called sumo-bots, where the goal is to push your opponent's outside of a ring. The robots are on a white table with a black circle around the perimeter. There are three levels of complexity using different strategies.

The line follower sensor will be used to tell the color of the ground. The basic strategy is if the ground is white, then the robot will drive forward trying to find an opponent to push. If the bot runs into the black line, it will back up and turn around. You can buy a BattleBot table from Robot Academy online at <https://www.robot-academy.com/schools>

The program for Battlebot utilizes other types of blocks such as LED, which controls the 5 LED lights on the bottom of the mBot2, Display, which controls the screen on the CyberPi, and the Operator type blocks, which perform mathematical calculations and logical comparisons to allow the robot to process data. The last new sections are the Quad RGB Sensor type block that analyses data from the color sensors on the robot and the Ultrasonic Sensor 2 type block. The Quad RGB sensors are used in this program to sense when the robot reaches the end of the arena, so it can move away from the border and stay in play. The Ultrasonic Sensor 2 uses ultrasonic waves to sense the surroundings of the robot by using ultrasonic waves. We use these sensors in the program to sense when there is an opponent in front of the mBot2 so it can increase the speed to charge at the opponent. The following image contains the program that we use to battle with the mBot2 robots. The program is used in tandem with LEGO shield creations that are created by the campers to effectively battle.

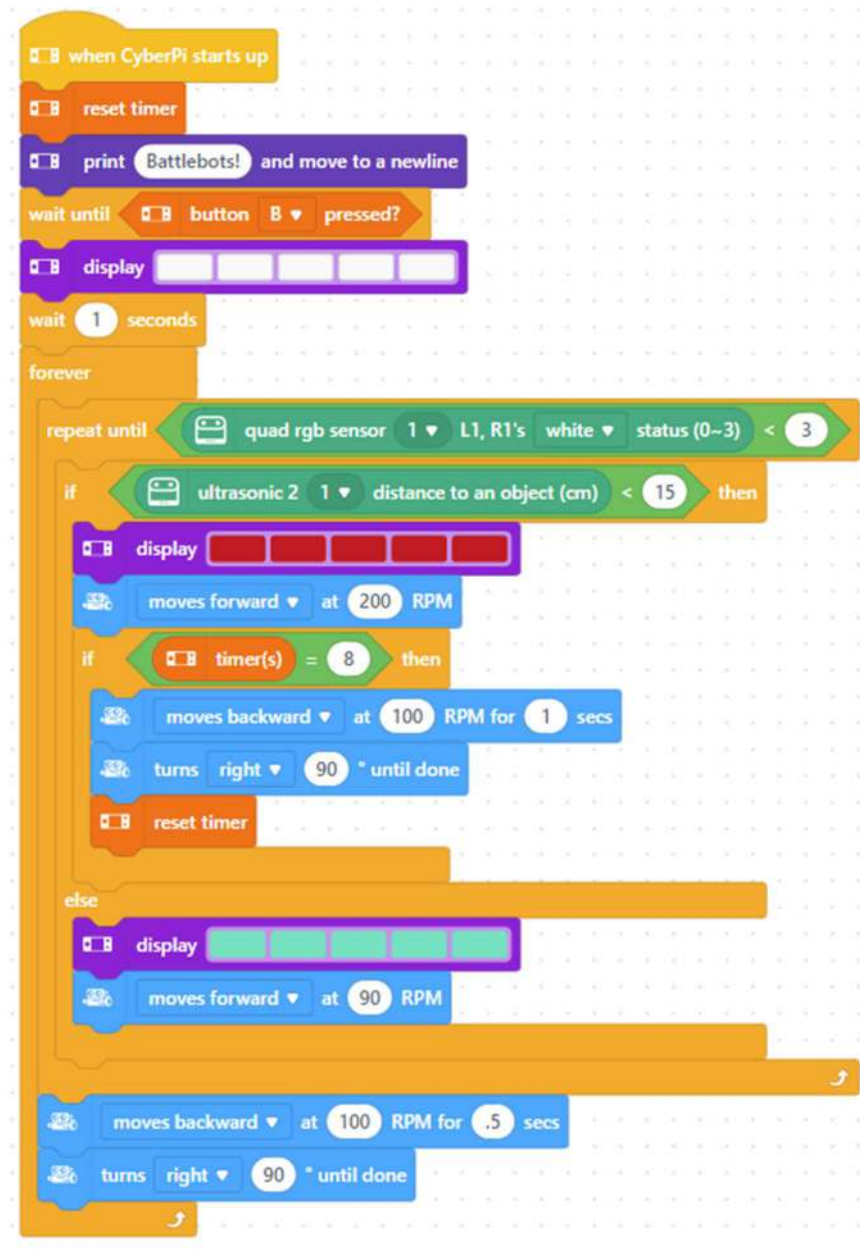


Figure 2 BattleBot Program



4. AI with mBot2

We explored the use of AI (Artificial Intelligence) blocks in Makeblock for programming the mBot2 robot. These AI blocks open up a world of possibilities, enabling campers to integrate sophisticated, intelligent behaviors into their robotics projects. AI blocks in the mBlock programming environment are specialized blocks designed to incorporate artificial intelligence functionalities into the code. These blocks allow students to create programs that can recognize patterns, make decisions, and perform tasks that typically require human intelligence. Here are some key AI blocks in mBlock programming

A. Voice Recognition

Voice recognition blocks allow the mBot2 to understand and respond to voice commands. Students can create programs where the robot performs specific actions based on verbal instructions, enhancing interactivity and user experience. During camp, we used voice recognition to guide our robots through a maze.

B. Translation

The translation blocks in the AI block section allows the robot to translate phrases that are spoken into the microcontroller.

Here is a screenshot of the AI interface within the CyberPi blocks that are available in mBlock. It is important to know that to access all of the AI blocks, it is necessary to sign into the with a mBlock account.



Figure 3 AI Blocks



5. Video Game Design

During this camp, you child had the opportunity to design, program, and test their own video game using the mBot2 CyberPi as a versatile controller. This project introduced them to the basics of game design and allowed them to apply advanced programming concepts in a fun and interactive way. We instructed the campers on several different games, but the following section will use the program for The Apple Game, in which the players are collecting falling apples.

The fundamentals of video games include defining objectives, creating characters, and developing storylines. Campers explored how to make games engaging and challenging by incorporating levels, scores, and interactive elements. Using the mBlock programming environment, students used block-based coding to create game logic, control movements using the CyberPi as a controller, and manage interactions between the robot and game environment. This approach makes complex programming concepts accessible and understandable for beginners.

The first step of the video game design with the mBot2 is integrating the CyberPi with the software to be used as a controller. The following screenshot is the code used initialize the CyberPi, which is encoded in the devices section with CyberPi selected as the device.



Figure 4 CyberPi Initialization Program

The interface of the video game is then encoded in the sprites section of mBlock. Characters, props, and text are made by creating new sprites which all have unique code embedded within them. The first sprite created for The Apple Game was an apple, which continuously fell from the top of the screen.

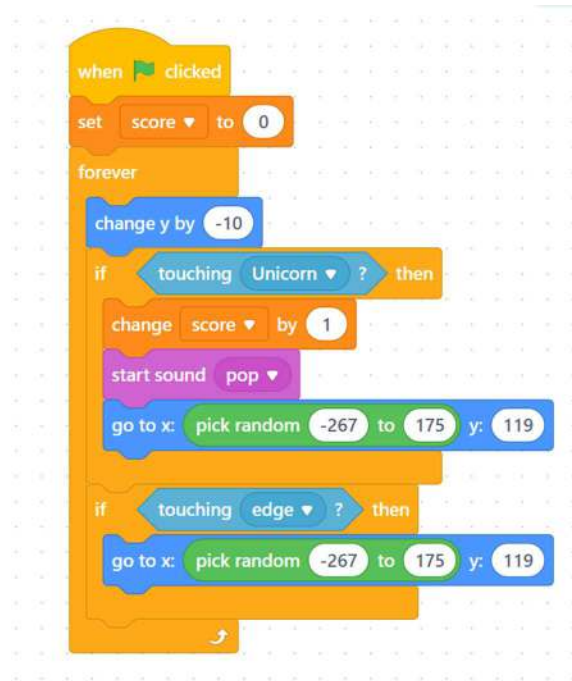


Figure 5 Apple Sprite Program

The next sprite created was the character, who was controlled by the CyberPi. This character would gain points by catching the falling apples. The movement in the x-plane of the character is controlled by the tilt of the CyberPi.

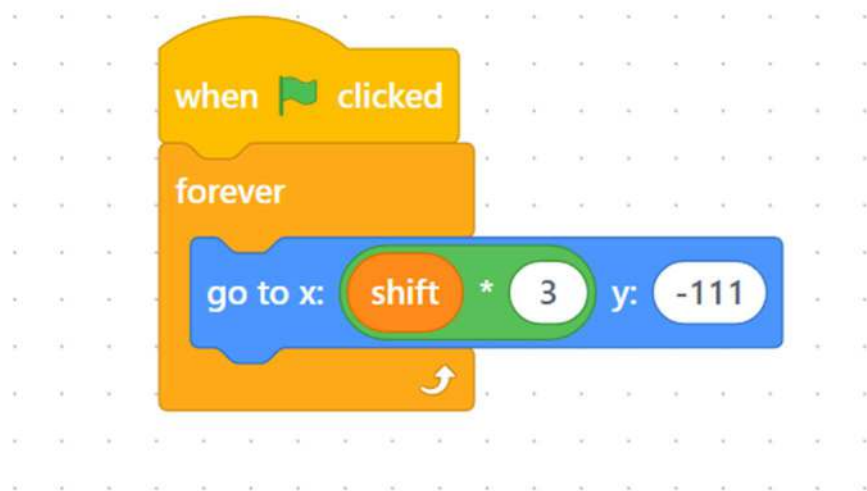


Figure 6 Character Sprite Program

The two following programs is the code for the WIN and LOSE sprites, which show depending on the score that the player achieves by the time that the timer has gone out. If the score is



greater than 30, than the WIN sprite appears, otherwise, the LOSE sprite appears. These sprites appear for 1 second at the end of the game, and then are hidden so that the game can be restarted.

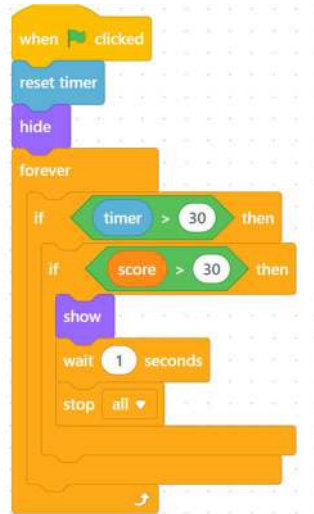


Figure 7 WIN Sprite Program

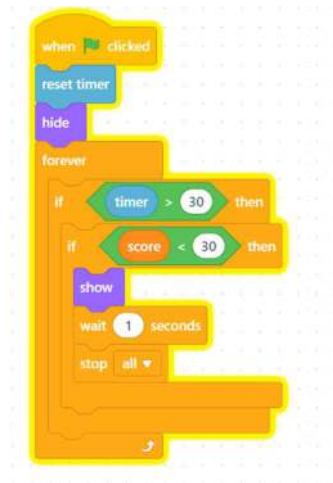


Figure 8 LOSE Sprite Program



6. Troubleshooting Guide

A. Installation

Download and use mBlock across different devices. Supports Windows, Mac, Linux, Chromebook, iOS, and Android. The mBlock software is free and available for download at:

<https://mblock.cc/pages/downloads>

B. Robot Won't Turn On

Causes would be low battery or loose connections. To solve this problem, check the batteries to ensure they are properly installed and charged. Replace or recharge batteries if necessary.

C. Robot Not Responding to Commands

Causes could be incorrect programming or communication issues between the robot and the controller. Solutions would be to review the code, double check the programming for any errors or missing blocks. Ensure that the code logic is correct and that all commands are properly defined. Check the connection to ensure that the CyberPi is properly connected to the mBot2.

D. Motors Not Working

Causes could be motor connections are loose or fault motors or drivers. Inspect the motor connections to verify that the motor wires are securely connected to the motor ports on the robot. Test the motors using the mBlock software to run a simple motor test program.

E. Sensors Not Responding

Causes could be that sensor connections are loose or the sensors could be faulty. Check the sensor connections to ensure all sensor cables are properly connected to the correct ports.

F. Additional Tips

- Consult the User Manual: The mBot2 user manual provides detailed instructions and troubleshooting tips for common issues. Refer to it for additional guidance.
- Visit the Makeblock Community: The Makeblock online community is a valuable resource for troubleshooting tips and advice from other users. You can find forums, tutorials, and FAQs to help resolve issues.
- Contact Support: If you encounter a problem that you cannot resolve, don't hesitate to contact Makeblock customer support for assistance. They can provide expert guidance and solutions.



7. Next Steps

- In-Person at Dublin, Upper Arlington, Powell, Reynoldsburg, and Grove City Parks and Rec Centers.
 - Arduino LEGO Robot Building and Programming
 - Arduino LEGO Robot Advanced Programming – this is perfect if you bought a robot and want to learn more advanced capabilities.
- Virtual Options:
 - Private 1-hour lessons or semi-private lessons with the Robot Geniuses at camp like Keaton and Ryan. Most lessons are weekly. Great help with troubleshooting or programming with your child's specific robot they designed.
 - We highly recommend this option for campers that bought the robot so that your child can continue to learn to build and program epic robots.
 - Register at <https://www.robot-academy.com/online-lego-robot-camp>
 - Please let me know if you are having any trouble beyond the troubleshooting guide and we can help. We have a support team for basic questions about installing the software and getting it working.
- BattleBot Birthday Parties or Private Events. Earn Girl Scout STEM Badges, Boy Scout and Cub Scout Robotics Badges with a LEGO Robot BattleBot workshop.

Contact gail@robot-academy.com for info.

Go LEGO,

Gail Vawter, Co-Founder and Owner

Robot Academy LLC

gail@robot-academy.com