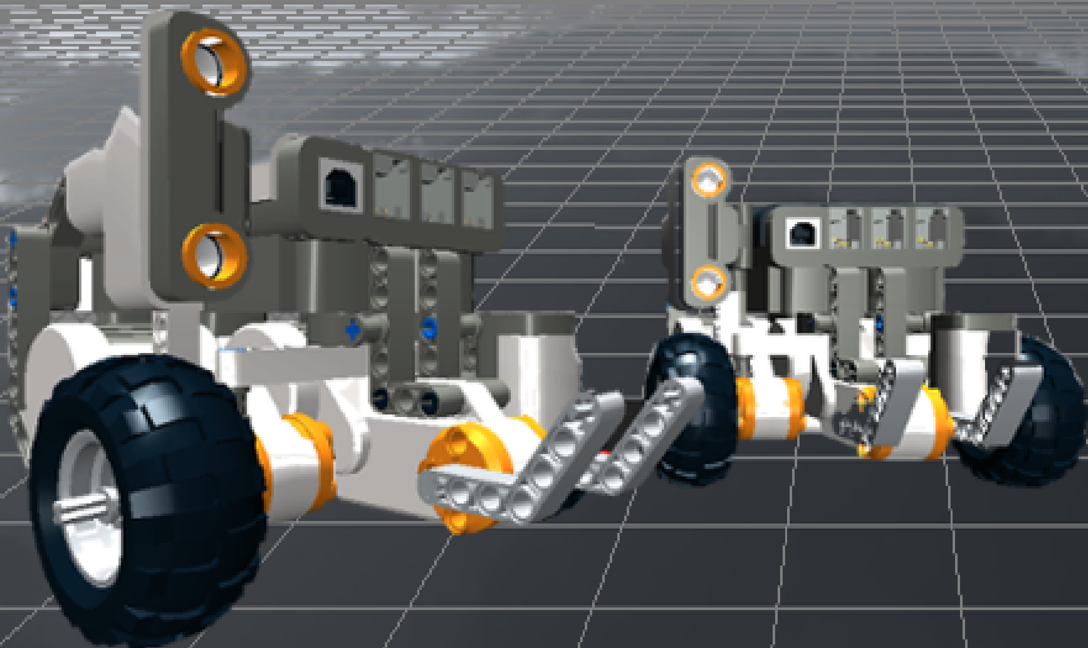
Virtual Robotic Toolkit

A Beginner's Guide



cogmation ROBOTICS

Ashish Singh, Chris Schulz
cogmation.com

December 2014

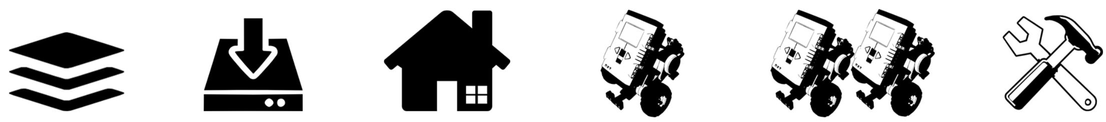
Preface

Designed for use with the LEGO® MINDSTORMS® NXT™ and EV3™, the Virtual Robotic Toolkit™ is a full on physics enabled simulator, that crams all the functionality and awesomeness of a real robot into a cute 1.3GB package.

The simulator is ideal for anyone who wants to experience the emotional highs and lows of designing and programming their own robot, but without the burden of ever needing space for testing, or running out of physical bricks.

This tool can be especially useful to those who are interested in teaching with robots, but who lack enough physical kits for each student in their class, and for robotics clubs who are looking for an excellent prototyping tool to help give them the edge over the competition.

This guide is split into six chapters, though chapters three through five are likely to be of the greatest interest to the reader.



Chapter 1 offers a brief introduction and history of the simulator. This chapter will also build the case for how simulated robotics can help to enhance and expand the LEGO® MINDSTORMS® experience. After reading this chapter, you should have a better understanding of what the simulator is and isn't, and how it can help you to become a better programmer.

Chapter 2 goes into the details of installing and setting up the system. It can be safely skipped, unless you happen to be a System Administrator or are really bored.

Chapter 3 is our first look at the simulator. This chapter will introduce the default projects that are available in the base installation of the Virtual Robotic Toolkit, and will suggest a course of study for learning how to use the sim. Details surrounding how to go about customizing the interface will also be presented.

Chapter 4 will methodically step through the first set of projects on the "Sim Basics" tab, and will gradually introduce the tools and concepts needed to go from driving the robot in a 3D environment, through writing programs to create autonomous behaviours, to discussing how to build a model from scratch. If you only read one chapter, make it this one!

Chapter 5 will introduce the more advanced multiplayer projects that exist on the Challenges tab.

Chapter 6 contains some potentially dangerous information on how to alter (and screw up) the default settings for the simulator. This chapter is provided only to give a sense of completion to the documentation, and is not for the faint of heart.

It is important to read the chapters in order - the guide is not that big, after all. Be sure to carefully read the examples, because a lot of the information is in the examples placed throughout the book.

If you have ideas for something to be added, removed or altered in this document, please let me know. I am especially interested in feedback from robotics novices about which bits of this intro are easy to understand and which could be explained better.

Chris Schulz <support@cogmation.com>

Cogmation Robotics
554 Pembina Hwy - second floor
Winnipeg, Manitoba - CANADA
R3M 2M4

Sincere thanks to Andrew Winton, Eric Himbeault, Connel for their great help and invaluable suggestions.

Contents

1. Introduction

This chapter presents a brief history of robotSim™, including a discussion on the many benefits of using simulated robots. The chapter will conclude by listing the new features that have been added to this release (version 2.7).

1.1 History of the Simulator

The Virtual Robotics Toolkit began its life almost 9 years ago, as a tool for researchers and engineers. Back then the idea was simple, before investing time and money in building an expensive real robot, perhaps it would be advantageous to have some way of testing on a digital prototype. As a result, the Cogmation simulator was born, and its earliest versions targeted expensive robots used primarily in academia, such as the Aldebaran® Nao™ and the iRobot® Packbot™.

Of course, creating a great design tool was just part of the solution. What we had really hoped for, was to use the simulator to teach robotics itself; and if that were the goal, then why not work with a robot specifically designed for the beginner? Why not design the simulator to work with a LEGO® MINDSTORMS®?

1.2 Why virtual robotics?

There are many benefits to adding simulation capability to the MINDSTORMS® experience, but a few of the more compelling reasons are given below:

- Perhaps the most obvious benefit, is that by using a simulated robot we can write programs to work with the robot, even if it isn't physically available. For example, your robotics club might only have a single MINDSTORMS® robot, but the whole team could be working and programming on a digital version of that same competition robot from home.
- When it comes to environments, simulation offers a much greater variety of challenges than would ever be physically possible. Although it might be too expensive to test our robot in the near weightless of space, we could easily simulate it.
- With virtual robotics, you never lose bricks, and it's always easy to find pieces

when you need them. By working in a completely digital space, we can build our model once and then keep it forever. Making it much easier to compete and share our robot with others around the world.

1.3 Video game vs simulation – what the simulator is and isn't

The distinction between simulation and video game is not always easily made, but it's a point worth considering. While both types of software allow for the design, creation, and manipulation of 3D objects, at it's core, simulations are about things (or systems) and how they behave. Most video games on the other hand, are about having fun.

The software you see before you is a complex, real-time, physics enabled simulator, it is not a video game. Although it is possible to have a fun time using it, it is a different kind of fun. It's the kind of fun that comes from inventing something cool, seeing it spring to life, and sharing it with others.

Despite this important difference, the simulator as a modelling tool can be incredibly effective at building models to compete in the kinds of complex games that are typical of FIRST Robotics challenges.

1.4 What's new in this version (2.7)

Some of the major new features that have been added in this release include:

- Support for the LEGO® MINDSTORMS® EV3™ programming environment
- The ability to import LDraw models into the simulator.
- Simplified user interface.
- Updated sample projects.

2. Installation and Setup

The Virtual Robotics Toolkit can be easily installed by following the steps in the installation wizard. The first time you launch the program, you'll be asked to either begin your trial or register the product.

To activate the product, ensure that your computer is connected to the internet, and enter the product registration key that you were given. If installing the software on a school network, a firewall exception to the url: `http://activations.cogmation.com` may have to be made.

2.1 Command line switches

The following is a table of available switches that can be used, should you prefer to install the software from the Windows command line.

Table 2.1: Command Line Switches

Switch	**Description**
-q, quiet, -s, -silent	silent install
-passive	progress bar only install
-norestart	suppress any restarts
-forcerestart	restart no matter what
-promptrestart	prompt if a restart is required (default)
-layout	create a local image of the bootstrapper (i.e. download files so that they can be burned to DVD)
-l, -log	log to a specific file
-uninstall	uninstall
-repair	repair (or install if not installed)
-package, -update	install (default if no -uninstall or -repair)

2.2 System Requirements

This Robotics Toolkit can be installed on any of the latest technology computing machines, however, we assume that it might be important for you to know the minimum and recommended technical specifications.

2.2.1 Available platforms

As of December 2014, the only operating system that the sim will run on is Microsoft Windows 7 or better. We anticipate that a Macintosh version of the simulator will be available sometime in 2015, please visit cogmation.com for more information.

2.2.2 Minimum system requirements

- Intel Core Duo or better
- 2 GB of RAM
- 1.3 GB of hard disk space
- Windows 7 or better

2.2.3 Recommended system requirements

- Intel iSeries, i3 or better
- 4 GB of RAM
- 1.3 GB of hard disk space
- Dedicated graphics card
- Windows 7 or better

2.3 Supported Robotics Platforms

2.3.1 LEGO® MINDSTORMS® NXT™ - (2nd generation robot)

The following is a list of NXT sensors that are supported in the simulator.

- Touch sensor
- Ultrasonic sensor
- Light / Color sensor
- HiTechnic Compass sensor
- HiTechnic IR sensor

Figure 2.1: NXT Intelligent Brick

2.3.2 LEGO® MINDSTORMS® EV3™ - (3rd generation robot)

The following is a list of EV3 sensors that are supported in the simulator.

- Touch sensor
- Ultrasonic sensor
- Color sensor
- HiTechnic Compass sensor
- HiTechnic IR sensor

Figure 2.2: EV3 Intelligent Brick

2.4 Additional Resources

2.4.1 About the LEGO® programming environment

The Virtual Robotics Toolkit is designed to work with both NXT and EV3 programming environments. If you're new to MINDSTORMS® programming, the language itself differs considerably from traditional programming syntax. Unlike other languages, MINDSTORMS® robots are programmed by stringing together sequences of modules to create new behaviours.

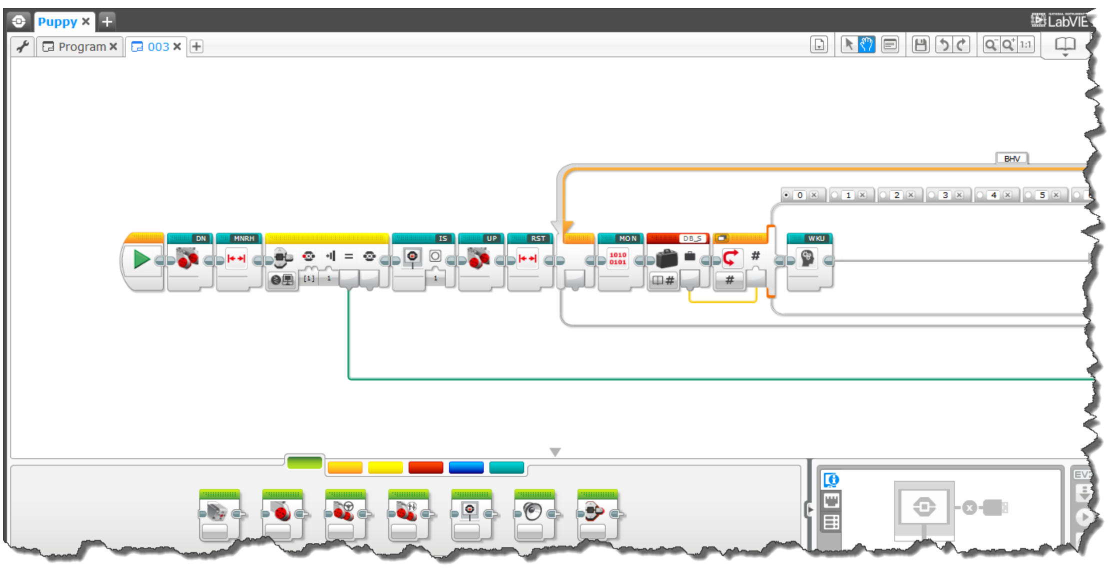


Figure 2.3: Sample EV3 Code

At the time of this writing, the LEGO® supplied programming environments for both the older second generation NXT, and the newer EV3 bricks, could both be freely downloaded from the official MINDSTORMS® community website.

`http://www.lego.com/en-us/MINDSTORMS/downloads`

While learning to use the MINDSTORMS® programming environment is outside the scope of this manual, there are a number of great resources on the web to help you get started. The programming environment itself also includes a number of tutorials built into the software. Here are some external resources that may be of some interest to the beginner.

NXT Resources: NXT Programs.com, (`http://www.nxtprograms.com`), has tons of great projects and programming tutorials for both the retail and education versions of the older second generation robot. I learned to program my first MINDSTORMS® robot by doing many of these same projects.

EV3 Resources: STEMcentric EV3 tutorials, (`http://www.stemcentric.com/ev3-tutorial/`), is a collection of screencasts that will outline the basics of writing programs for LEGO®'s third generation robot.

Blogs and social media: The NXT step,(`http://www.thenxtstep.com`), is the pre-eminent blog on all things MINDSTORMS® robots. FLL Casts.com, (http://www.fllcasts.com), is a subscription site, but well worth the money with many excellent webinars filled with helpful hints for FIRST® LEGO® League teams.

2.4.2 About modelling using LEGO® Digital Designer™

LEGO® Digital Designer™ (or LDD, as it is more commonly known) is a freely available CAD tool. If you have any interest at all in designing your own digital robots, you will want to seriously consider becoming acquainted with this design tool.

What makes modelling with LDD so easy, is that the software closely mirrors the real life process of building using physical bricks, by highlighting available connection points ('snaps') between model elements. Once your model is built, LDD can produce both a bill of materials as well as a set of building instructions that can be used to construct the physical robot.

LEGO Digital Designer can be downloaded (free of cost) from the link below:

`http://ldd.lego.com/en-us/download/`

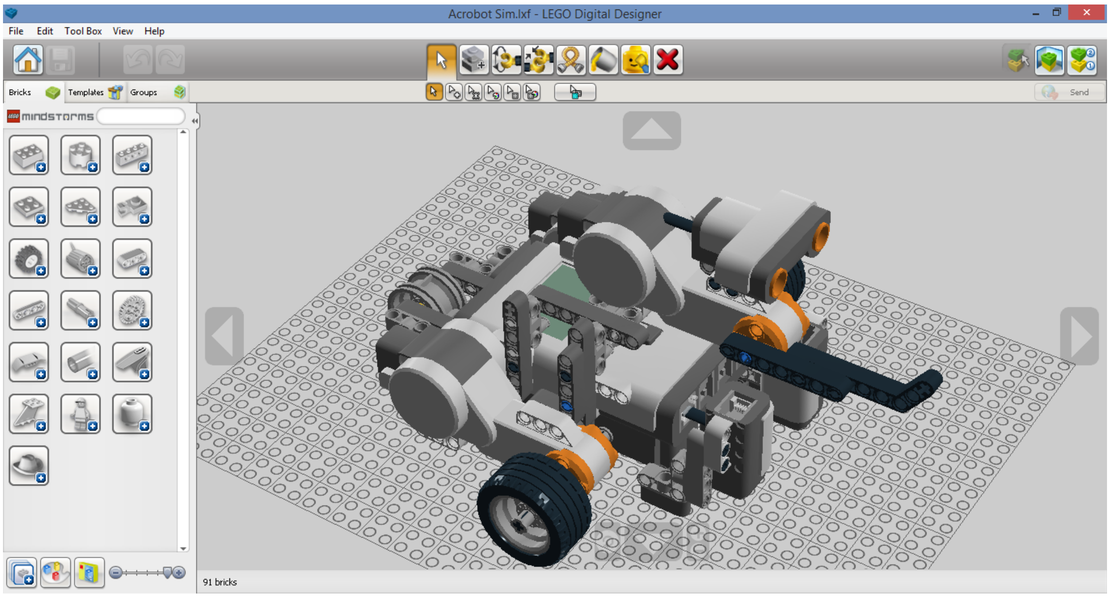


Figure 2.4: Sample 3D model built using LEGO Digital Designer

2.4.3 LDraw

In addition to LEGO Digital Designer, you may also want to consider installing the LDraw "All-in-One" parts library. LDraw is one of the many file formats supported by LEGO Digital designer, and is probably the most popular file format for creating and sharing 3D LEGO models between CAD programs.

The Virtual Robotics Toolkit contains an import wizard which can be used to bring LDraw files into the simulator. While it is not necessary to install this library, having it installed will afford many more options for the types of bricks that can be brought into the simulator.

The LDraw library can be downloaded (free of cost) and installed from the official website listed in the link below:

`http://www.ldraw.org/help/getting-started.html`

After installing the 'All-in-One' library, you will also want to update LEGO Digital Designer so that you can export your models to the sim. This can be achieved by clicking on **Help** menu and selecting **Patch LEGO Digital Designer**.

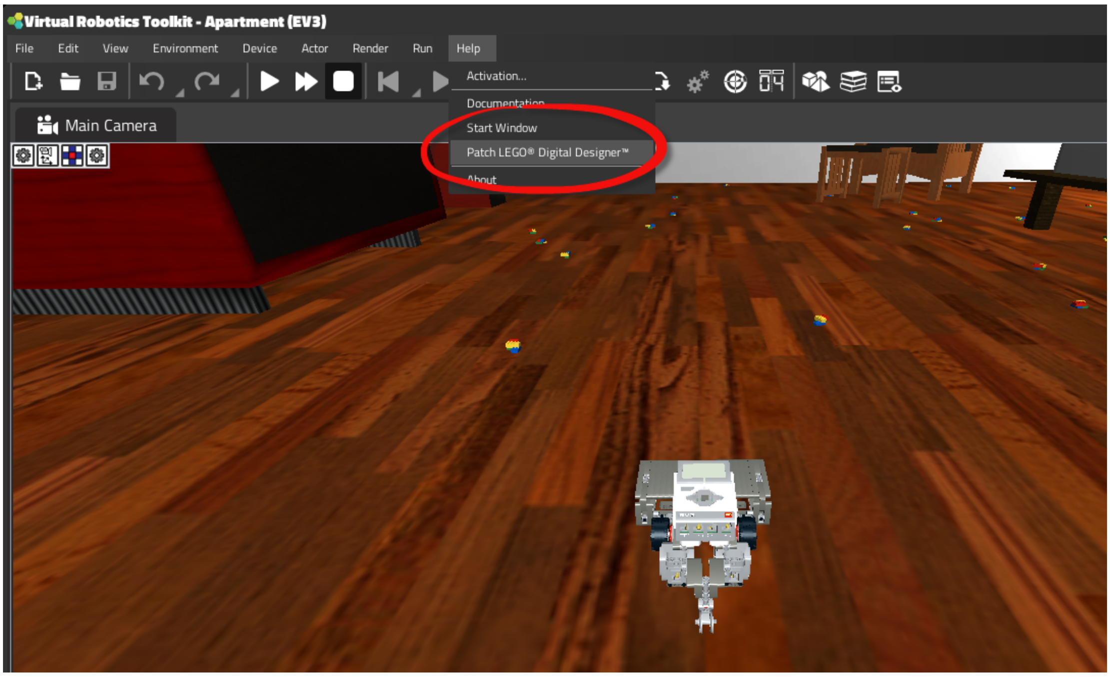


Figure 2.5: Applying the patch for LEGO Digital Designer

2.5 Summary

In this chapter we outlined the process for installing and setting up the Virtual Robotics Toolkit on to a computer. While the software usually installs without issue, in school computer labs, it is occasionally the case that an exception must be made to a network firewall to allow the Activation Manager to reach the Cogmation licensing server.
Like most applications, the Toolkit runs best on a new computer with lots of memory and a multi core processor, and even better when that computer has a dedicated graphics card. With the fun stuff out of the way, we can look forward to exploring the projects that are available to us.

3. Welcome Screen

In this chapter, we will take our first look at the default projects that are included in the base installation of the Virtual Robotics Toolkit, and will recommend a course of study. We will conclude this chapter by demonstrating how the simulator interface can be customized to best suit your needs.

3.1 Projects

As you may have already noticed, upon launching the simulator, you are greeted by the Welcome Screen. In the base installation, this window is divided into two tabs.

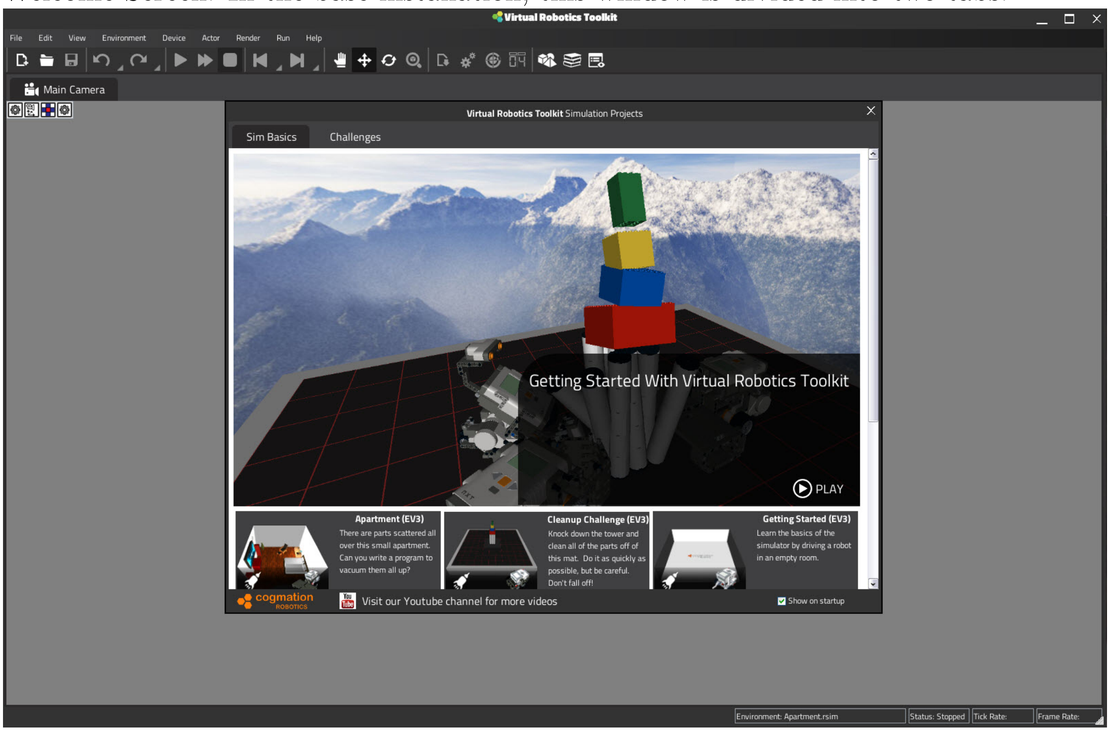


Figure 3.1: Welcome Screen

The first tab, "Sim Basics," contains four projects to help you get started. A brief overview of each project as well as the recommended sequence for tackling each challenge is provided below:

Figure 3.2: SimBasics Tab

- **Getting Started** is nothing more than an empty room with a robot. The goal of this distraction free environment, is to learn the basics of starting and stopping the simulator, and how to use the keyboard controls to drive the robot.
- **Maze** is the next project you should practice with. The goal of this project is to learn how to connect the MINDSTORMS® programming environment to the simulator.This will allow us to solve the maze by writing our own programs using MINDSTORMS® software and uploading it to the existing virtual robot in the simulator.
- **Clean-up Challenge** is the "build" challenge in this training series. While we can use the given robot that is provided to clear the playing surface, the real goal here is to introduce LEGO Digital Designer and demonstrate how we can use the LDraw import wizard to work with our own customized robots.
- **Apartment** is the last in the series of training projects. Here we can use the given robot along with its various attachments, and use our skills to program it to navigate around the room and vacuum the blocks that littered the floor.

Each of the above mentioned projects will be elaborated in the next chapter.
The second tab on the Welcome Screen, presents three additional multi-robot "Challenges" where we can test our newly learned skills against another friend or autonomous robot.

Figure 3.3: Challenges Tab

- **Multiplayer Sweep** is a two robot variant of the Clean-up challenge. Here we can either use keyboard controls to compete head to head against another player, or alternatively load different control programs into multiple Intelligent Bricks.
- **Sumo** is a digital version of the popular robotics challenge, where two robots are programmed to push each other out of a ring. The last robot standing wins.
- **Soccer** is a simulated version of the World Robot Olympiad - GEN II soccer game. This is probably the most complex of all the challenges. Players will have to program their robot to work with the HiTechnic® sensors to locate the soccer ball and push it into the opposing goal.

These multi-robot projects will be discussed in further detail in Chapter 5.

3.2 Customizing the Interface

Since no two people work the same way, it might be worthwhile to explain how we can customize the simulator‘s interface, before starting our first project.
Perhaps the easiest and most effective way to customize the workspace is to click on the View menu and select either Clear or Restore windows.

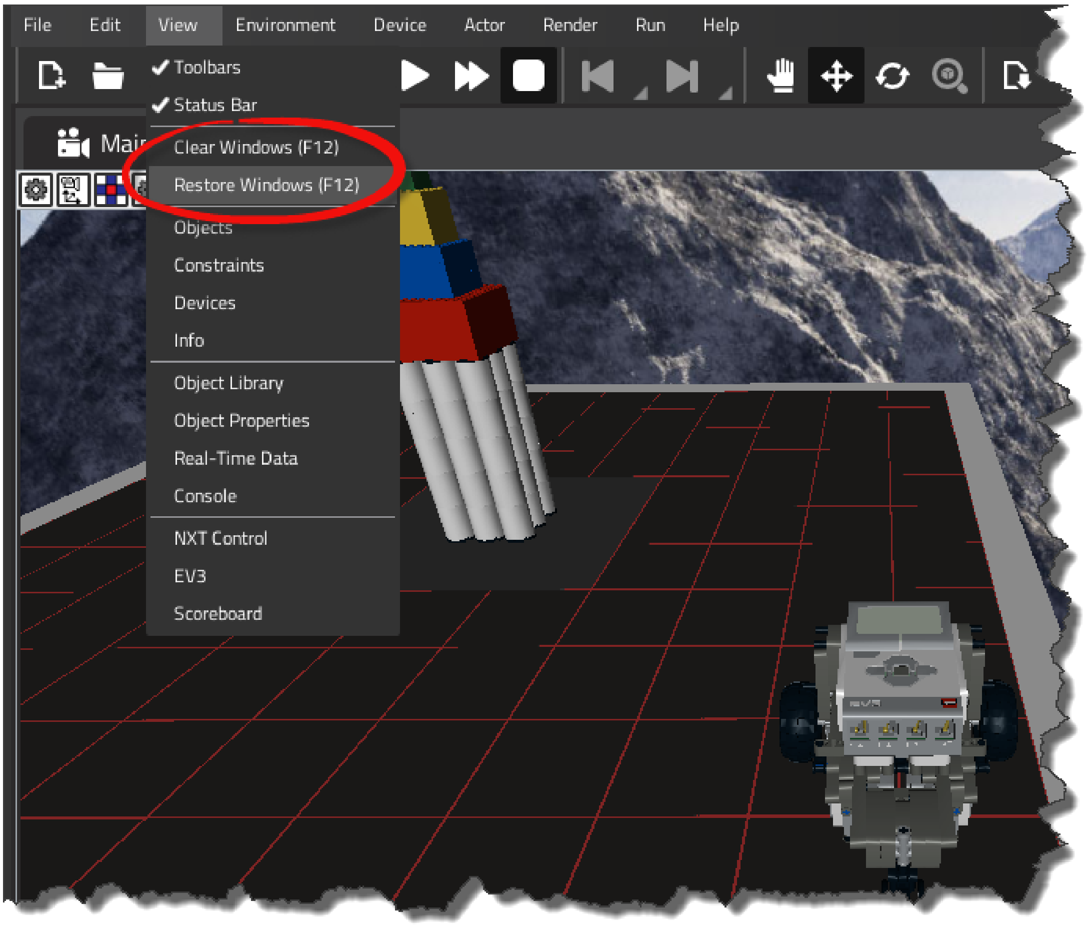


Figure 3.4: Clear / Restore windows

When Restore Windows is chosen, any windows that were previously docked will now reappear on the screen. Alternatively, choosing to Clear Windows from the View menu will as you might have guessed, maximize our available screen space by hiding all docked windows, and closing any floating panes.

We can toggle between these two commands by pressing the F12 keyboard shortcut.

If there are any additional windows that we wish to work with, we can easily add them to the screen by again clicking on the View menu and choosing the desired dialog.

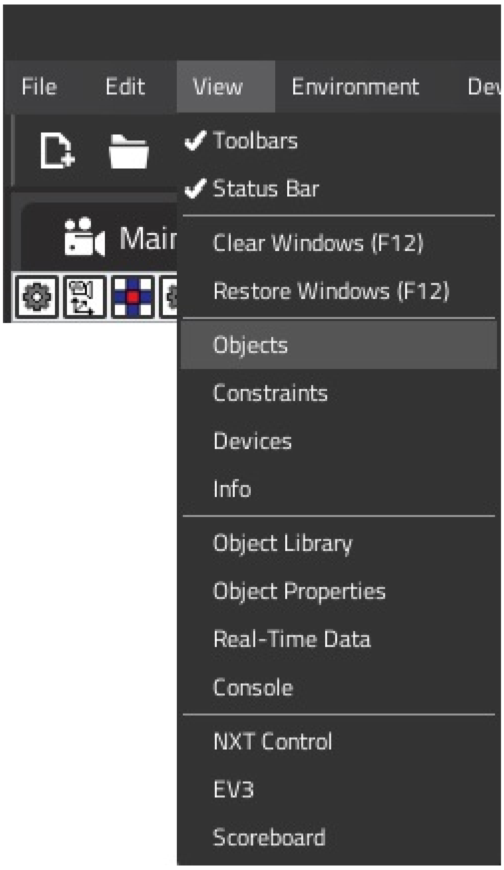


Figure 3.5: View Menu

When a new window is added, the default setting for the dialog is to float over the screen. To dock a window to a particular place, simply select the title bar of the window and drag and drop it to the desired location. As you drag the window, a list of available docking locations for the dialog will display on the screen.

If multiple windows are docked in the same location, they will be displayed as a series of tabs.

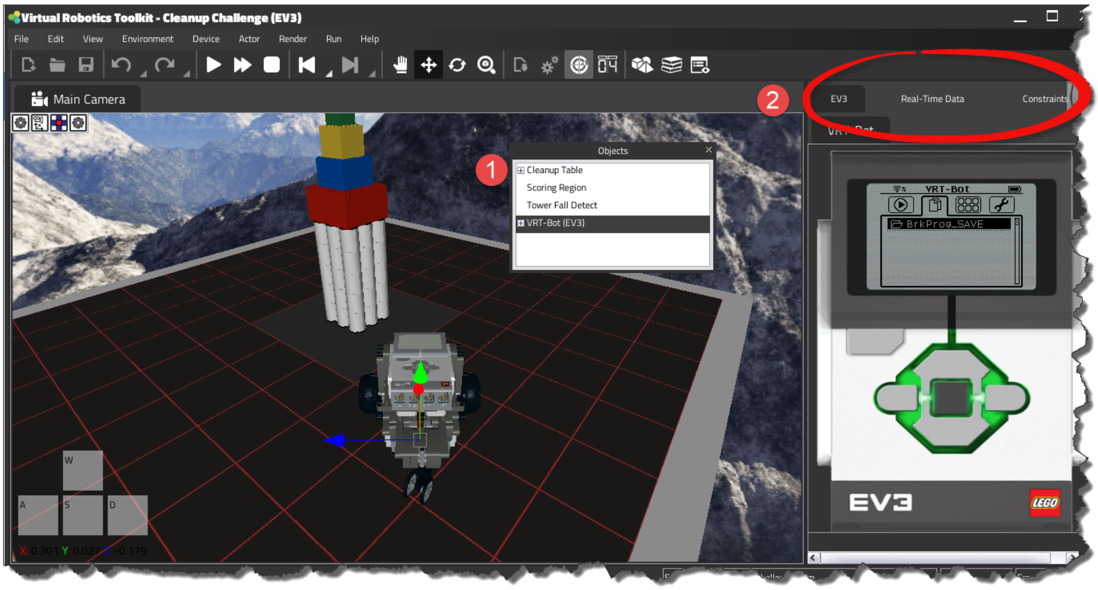


Figure 3.6: 1. Floating Window, 2. Multiple docked windows with tabs

Figure 3.7: Closing / Undocking windows by right clicking

To undock or close a window, right click on the title tab and choose either Close or Undock.

The first time the simulator is opened, the default setup is to have the Scoreboard and Objects window docked to the right hand side of the screen.

3.3 Summary

Knowing how the projects in the Sim Basics tab are organized, and tackling them in the correct sequence, is the key to building the foundation needed to approach the more interesting multi-robot challenges. By understanding how to customize the simulators interface, you can make these tasks much more enjoyable by creating a workspace that suits your needs.

4. Simulator Basics

In this chapter, we will methodically step through the first set of projects in the Sim Basics tab, and will gradually introduce the tools and concepts that are needed to design, test, and program our very own virtual robot.

4.1 Getting Started

4.1.1 Project Overview

Figure 4.1: In the **Getting Started project**, we will explore the basics of using the simulator. The goal here is to familiarize ourselves with the simulator controls and use them to pilot our robot through an empty space.

4.1.2 Opening a project

Within the Welcome Screen, find the Getting Started project and click on the tiled image to open it. Next, you will be asked to give your project a name. The simulator will first save the project to your Windows "My Documents" folder before opening.

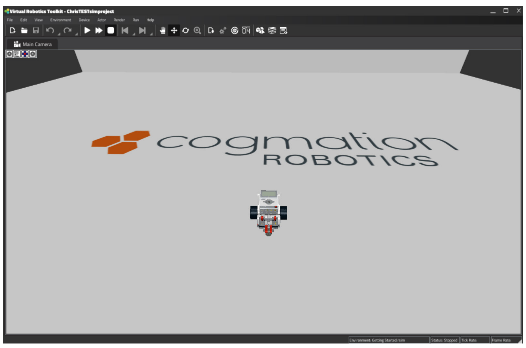
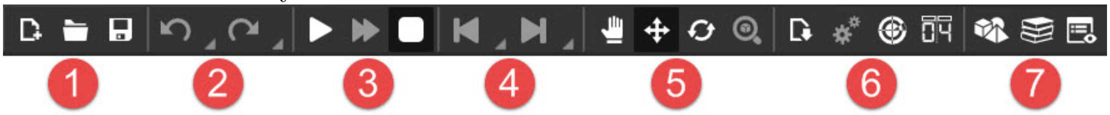
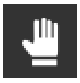


Figure 4.2: With the project file open, we can see that the simulator is divided into two areas. A large **Main Camera** area, also known as **model space**, where we can interact with the 3D environment, and a **Toolbar** that sits at the top of the screen and provides quick access to various simulator features and resources.

The Toolbar is further divided into seven panes, where each pane contains grouped shortcuts to commonly used features.



1. File management
    - Shortcuts to the new, open and save dialogs.
2. Undo/Redo stack
    - Allows you to undo or redo an action on an object (e.g., object position), it also keeps a list of recent commands (e.g., add or delete object).
3. Simulator control
    - Is used to start, stop and pause the simulation.
4. Previous / Next states stack
    - Allows you to step through a series of saved simulator states. For example, while driving the robot along the walls of a room, we could "pause" the simulation as the robot rounds each corner. Clicking on the "backwards" or "forwards" buttons in the "Previous / Next states stack" will then either revert or advance the robot to different stages (or corners) in the simulation.
5. Manipulation tools
    - Contains shortcuts to commonly used tools for manipulating an object.

    Truck Camera, allows us to adjust the position of the main camera from a constant distance.

    Move Object, is the primary tool that is used for selecting objects in model space, and allows the user to reposition the model along the x,y,z axes.

    Rotate Object, is used to change the rotation of a selected object along the x,y,z axes.

    Find Object, will locate an object in model space when it is selected from the **Objects** window.
6. Robot tools
    - Contains shortcuts to commonly used features when working with a robot.

    Import Wizard, allows the user to import an LDraw (*.ldr) format robot into model space.

Manage Attachments, allows the user to add or remove attachments from the selected robot. If no robot is selected, this button will be disabled.

Realtime Data, allows the user to see the current readings from the robots' attached motors and sensors.

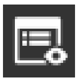
Scoreboard View, will open the projects' scoreboard.

7. Objects

   - Contains shortcuts for accessing objects and their properties.

Objects, lists the objects currently in model space.

Object Library, is a collection of 3D models that can be added to model space.



Object Properties, where we can adjust the physical attributes of a selected object.

4.1.3 Starting and stopping the simulator

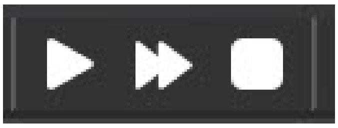


Figure 4.3: At this point, the **Toolbar** buttons that we are most interested in are those in the **Simulator control pane**.

As you may have guessed, the simulator operates in both an "on" and an "off" state.

In the "off" state, we can freely move and rotate objects in the model space.

In the "on" state, the simulators' physics engine is activated, making it much more difficult to manipulate objects since we must now overcome physical forces such as gravity.

4.1.4 Driving the robot using the keyboard keys

That said, we can only drive and program our robot when the simulator is running.

To turn the simulator "on", press the Play button and begin driving the robot using the W,A,S,D keyboard keys.

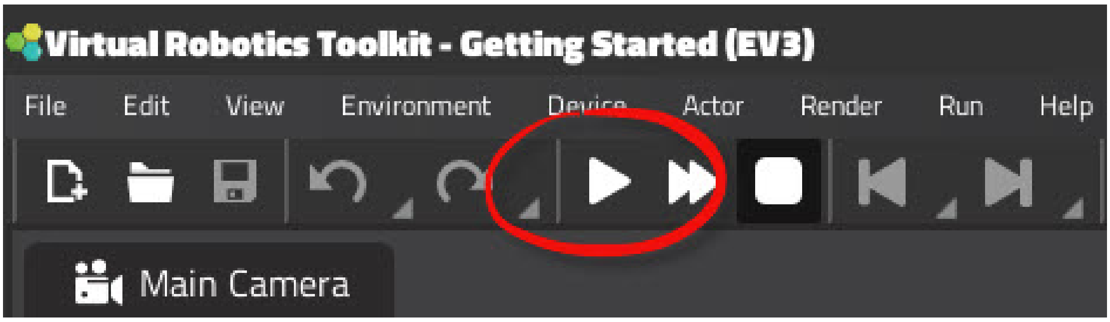
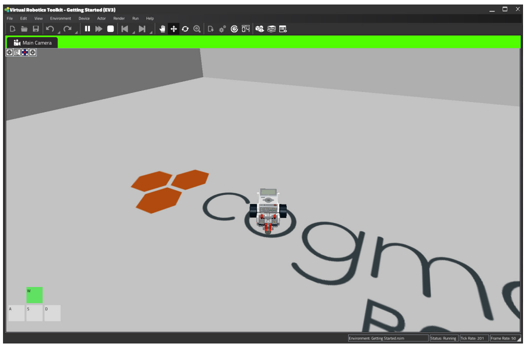


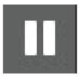
Figure 4.4

A green band will appear at the top of the screen to let you know that the simulator is "on". As each key is pressed, it will be highlighted in the on-screen display, and cause the robot to move in the specified direction.

Figure 4.5:
W = moves the robot forwards
S = moves the robot backwards
A = turns the robot left
D = turns the robot right



4.1.5 Resetting and restoring past simulator states

Take a few minutes now to experiment with driving the robot. Along the way, occasionally pause  the simulator to save your position to the Previous / Next states stack.

With the simulator paused, we can now retrace the path that the robot has driven by going back to a previously saved state. Click on the Previous and Next state buttons to experiment with this feature.

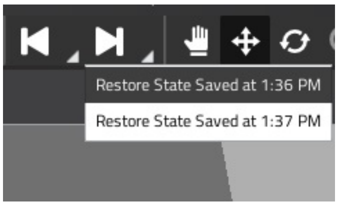


Figure 4.6: Restoring saved state

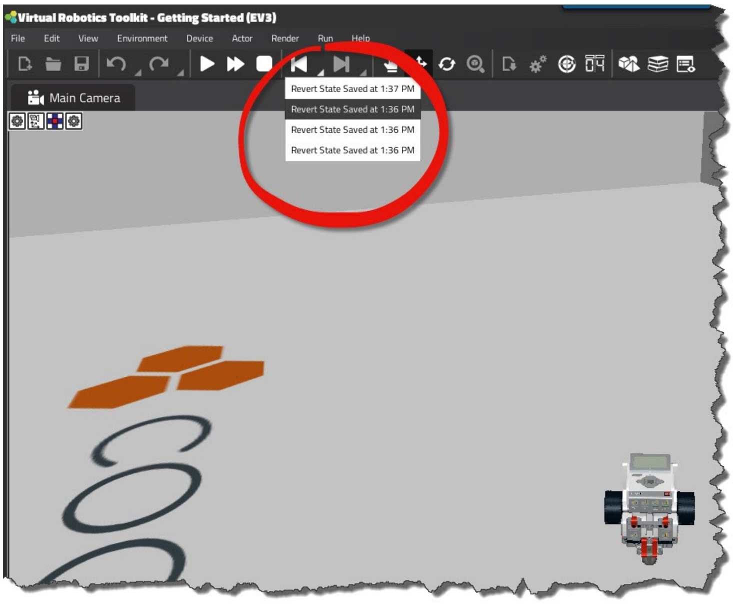
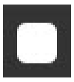
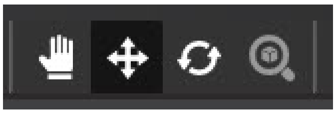


Figure 4.7: Restoring saved state

When finished, click the Stop  button on the Toolbar to end the simulation and restore the robot back to its original position.

4.1.6 Using the toolbar to manipulate objects



In addition to controlling the simulator, we can also use the Toolbar to control the placement and rotation of objects in model space.

Move Object tool, allows us to reposition an object in model space along a set of 3D (X,Y,Z) axes.

Rotate Object tool, allows us to rotate an object along a set of 3D (X,Y,Z) axes.

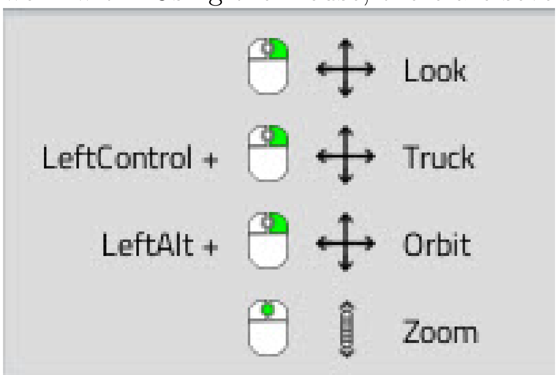
Find Object tool, which can be used to locate an object that was selected in the Objects window.

Take a moment now to adjust the placement and rotation of the robot, by selecting a button from the Toolbar, clicking on the robot, and then pulling on the axes.

Figure 4.8: Rotate objects

4.1.7 Moving around with the mouse

When manipulating objects in model space, half the battle is just finding a good view to work with. Using the mouse, there are several ways that we can adjust our view:



Look by right clicking on your mouse to look around.

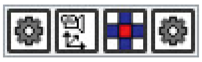
Truck by holding the left Control key + right clicking.

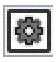
Orbit by holding the left Alt key + right clicking.

Zoom by using the middle mouse scroll wheel.

Alternatively, we can also use the keyboard to zoom in and out from an object by pressing the '+' and '-' keys.

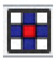
4.1.8 Camera tracking options





Near / far clip options, refer to the near and far plane of the viewing frustum, the region of space in the modelled world that is visible on screen. Anything closer to the eye than the near clipping distance isn't displayed (it's too close), and anything further away from the eye than the far clipping distance isn't displayed either (it's too far away).

Camera presets allow us to reorient the camera along a number of predefined perspectives, such as from the top of the object, the left of the object, the front of the object, etc.



Perspective / Orthographic view toggle, an orthographic view is typically used in engineering applications for viewing objects at a fixed depth. That is, as the camera zooms in and out, the objects in view will not become either bigger or smaller. This is in marked contrast to a perspective view, which is typically how we normally see the world, in which as we get closer to something it becomes bigger, and as we get further away from an object it appears smaller.

Camera tracking options, allow us to select an object in model space and to have the camera follow it as it moves.

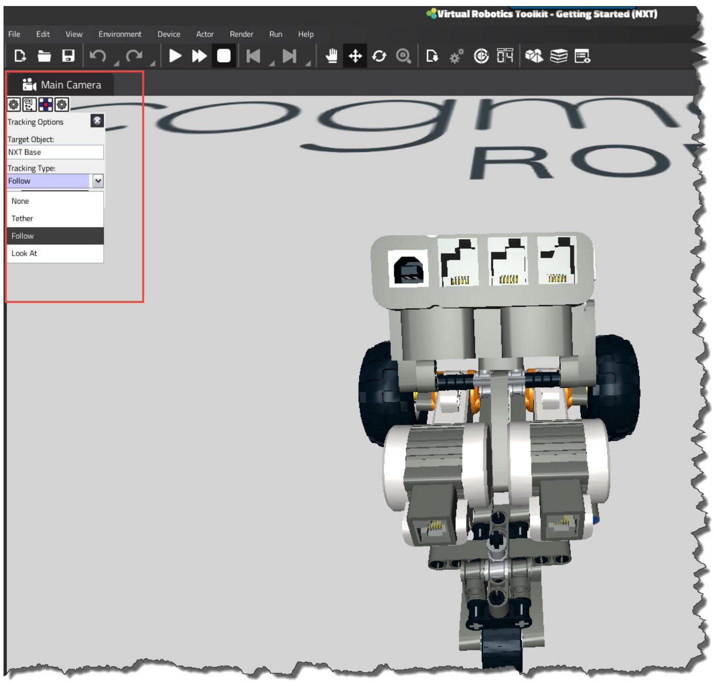
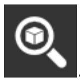


Figure 4.9: Camera tracking options

4.1.9 Finding objects

Should you lose sight of an object, you can easily locate it by following these steps:

1. Click on the **Objects** toolbar button .
2. Select the object you wish to locate from the list.
3. Click on the **Find Object** toolbar button .

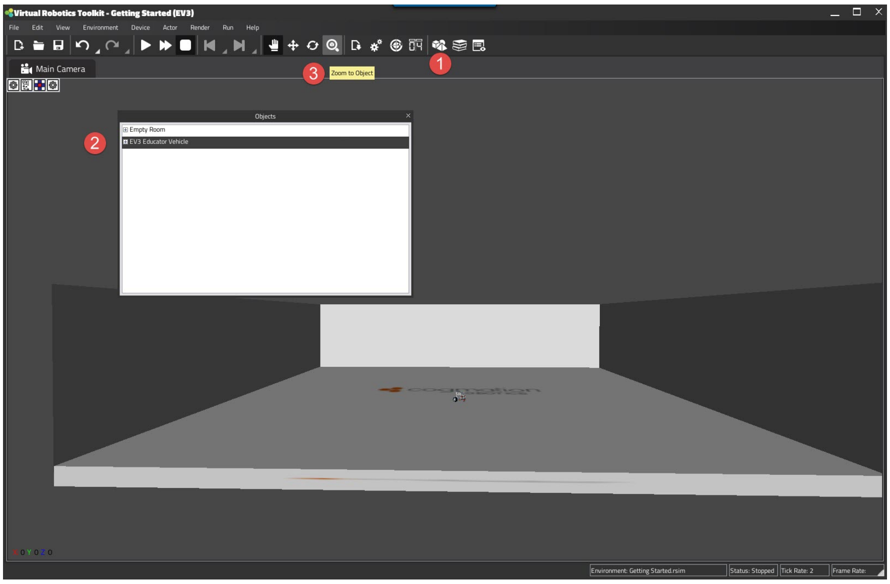
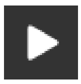


Figure 4.10: Find Object

4.1.10 Tracking the robot

If visualizers are turned on ('Render' menu → 'Visualizers'), we can use them to see the path our robot has travelled. To track the robot's movement, follow these steps:

1. Click on the **Objects** toolbar button. 
2. Select the robot from the list.
3. Right click to bring up a context menu for the object, choose "Track Movement".
4. Start the simulator, by pressing the **Play** button. 

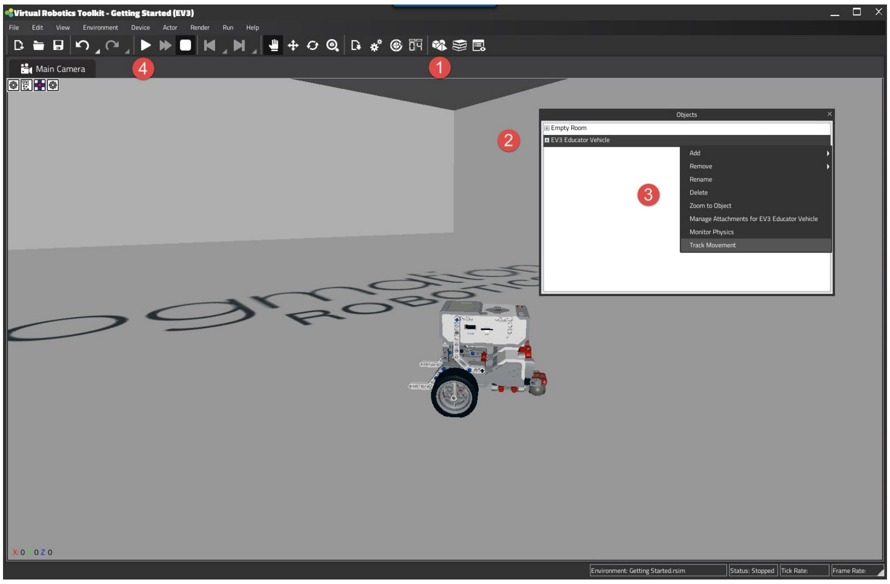


Figure 4.11: Tracking the robot

We'll note that with the simulator running we can now drive the robot using the W,A,S,D keyboard keys, and a blue line will be drawn behind the robot as it travels.

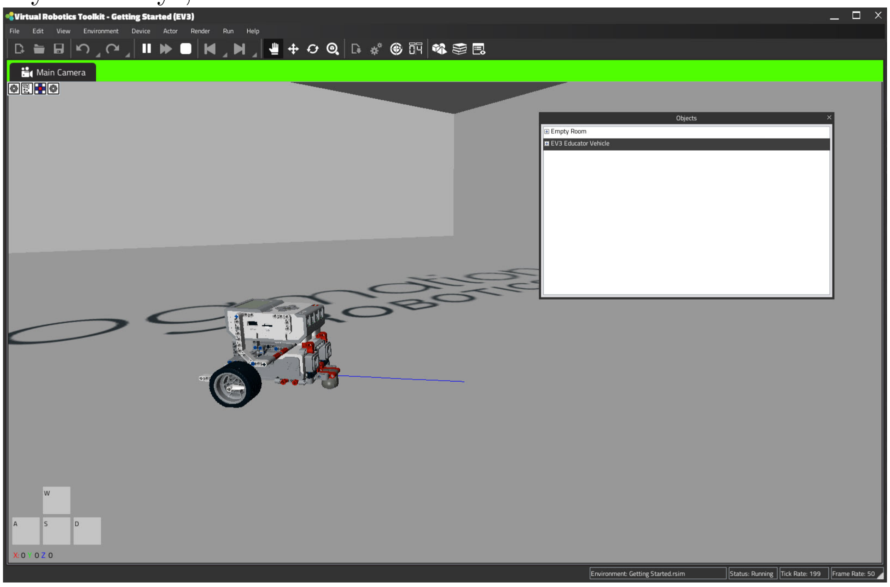
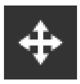


Figure 4.12: Robot leaving behind "Breadcrumb"

The "Breadcrumb" or track that the robot leaves behind, can be cleared by right clicking on the robot in the **Objects** window and choosing to "stop tracking" and then clicking again to "clear tracks".

4.1.11 Using attachments

Many of the sample project robots come with additional components that can be added. To see a listing of available attachments for the sample project robot, select the robot by either clicking on it in the **Objects** window, or by clicking the **Move**  button located on the toolbar, then clicking on the robot in the Main Camera.

With the robot selected, click on the **Attachments**  button located on the Toolbar.

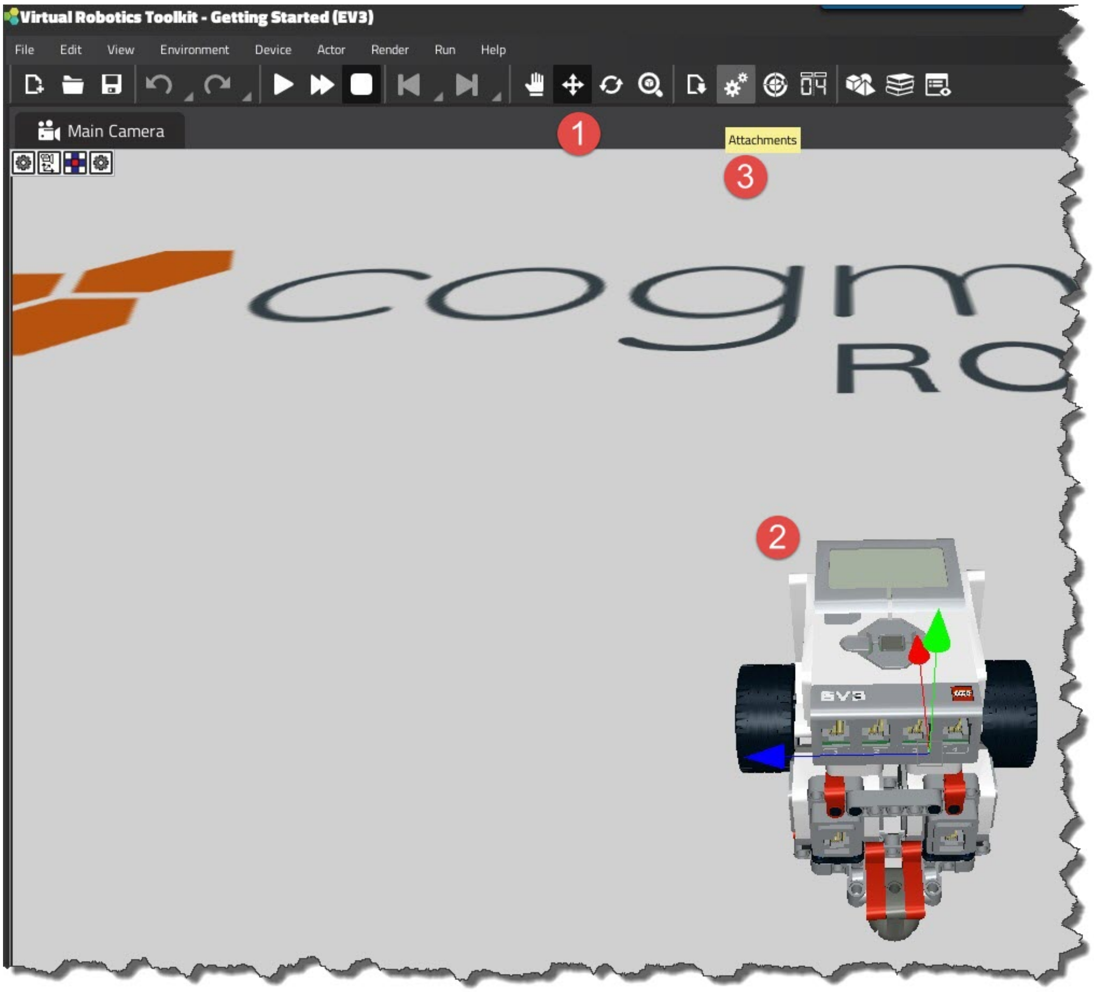


Figure 4.13: Adding attachments

After clicking on the **Attachments** button, a new window will appear.

To add a new attachment, select it from the list of "Available Attachments", choose a position on the robot, and then click "Add".

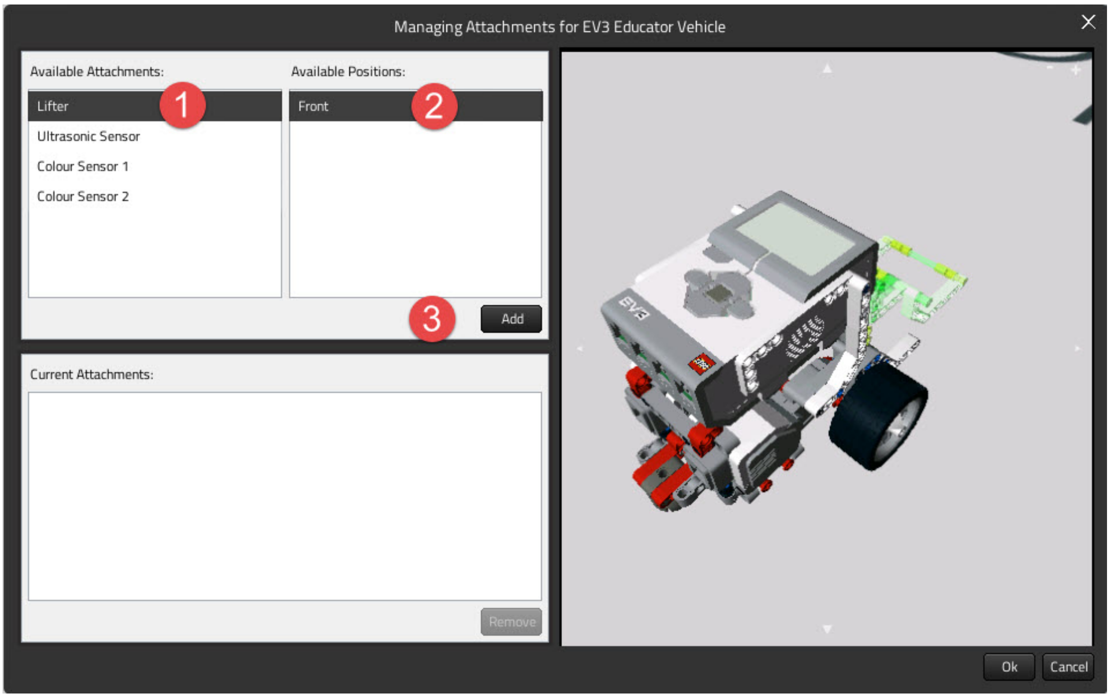


Figure 4.14: Adding an attachment

Similarly, you can remove an existing attachment by selecting it from the "Current Attachments" list, and clicking on the "Remove" button.

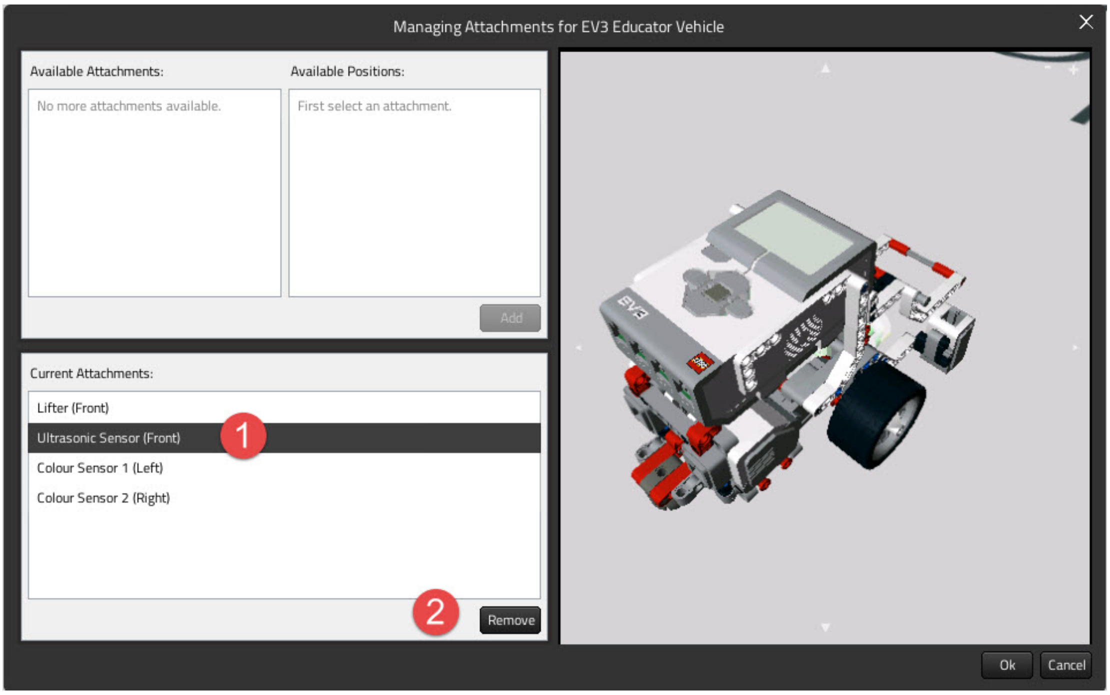


Figure 4.15: Removing an attachment

When finished, click the "Ok" button to save your changes.

4.1.12 Summary

In this section, we explored the basics of moving and manipulating a robot in three dimensional space. We also saw how we can take advantage of camera placement to make it much easier to work with the sample project robot.

Before going any further, become comfortable with the simulators' keyboard and mouse controls. We'll use them extensively in the section on programming that follows.

4.2 Programming

4.2.1 Programming overview

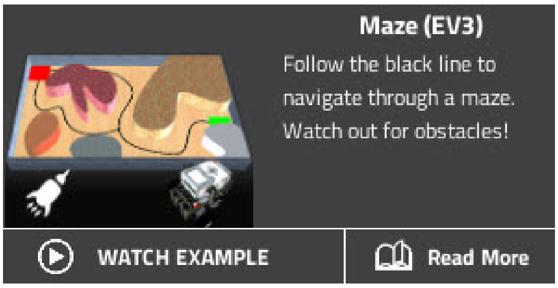


Figure 4.16: The Maze is where we'll learn how to connect the LEGO® MINDSTORMS® programming environment to the simulator.

This section will conclude by discussing how to work with sensors, and demonstrate how we can use the information that they provide to navigate through the maze.

4.2.2 Introducing the LEGO® MINDSTORMS® programming Environment

While a tutorial on MINDSTORMS® programming is outside of the scope of this document, there's a few basic things which we'll need to know in order to get started.

As was first mentioned in the second chapter, both programming environments for the MINDSTORMS® EV3 and NXT can be freely downloaded from LEGO® by clicking on the link here: `http://www.lego.com/en-us/MINDSTORMS/downloads`.

While the differences between the two programming environments is largely cosmetic, it's worth mentioning that the steps involved in connecting the LEGO® programming environment to the simulator will vary slightly between platforms.

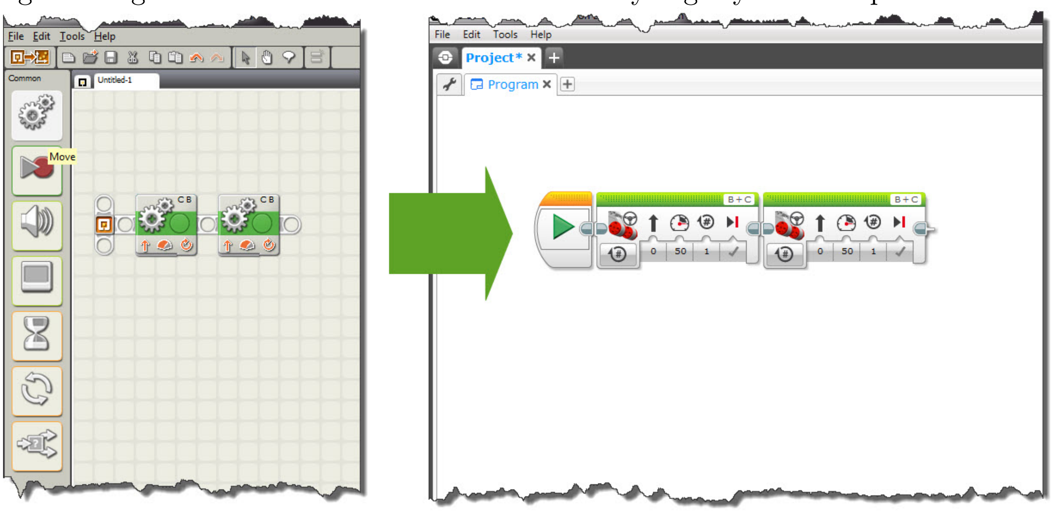


Figure 4.17: NXT vs EV3 programming environment

4.2.3 Running our first program in the simulator using NXT-G

1. Open the NXT-G programming environment, and create a new program called "MoveRobot". This simple program will do nothing more than use a single MOVE block to propel the robot forward for 1 motor rotation.

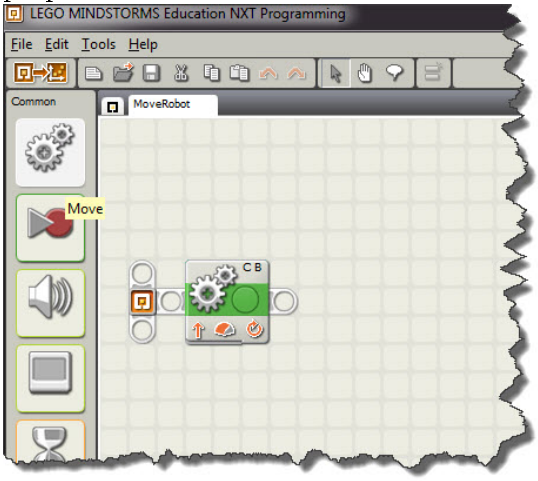
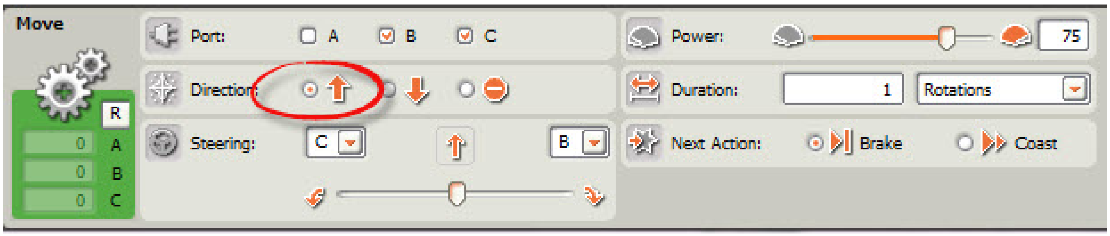


Figure 4.18: simple NXT program.



2. Click on the NXT-G Tools menu, and select "Download to File".

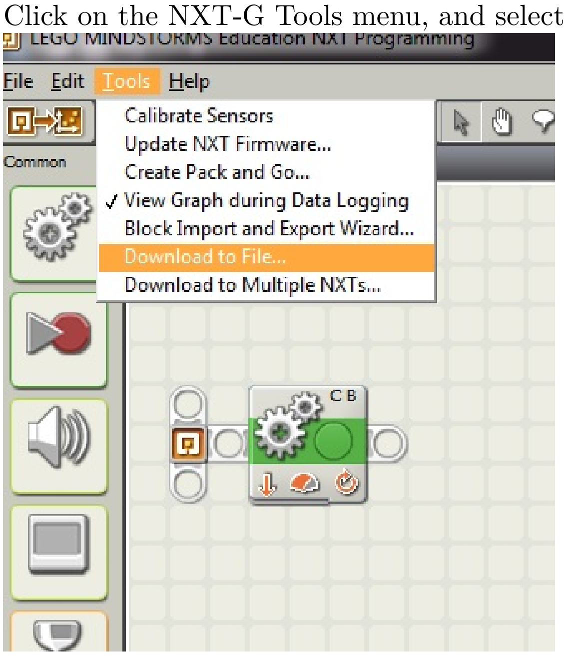


Figure 4.19: Saving the compiled NXT program to the desktop

3. A window will pop-up asking where you would like to save the compiled program file (.rxe). In this example, we will save our program to the desktop.

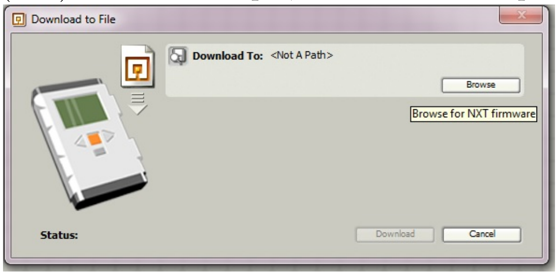


Figure 4.20: Download to file dialog

4. Open the simulator, if it isn't already opened, and select the Maze.
5. In the simulator, click on the "View" menu, and choose "NXT Control".

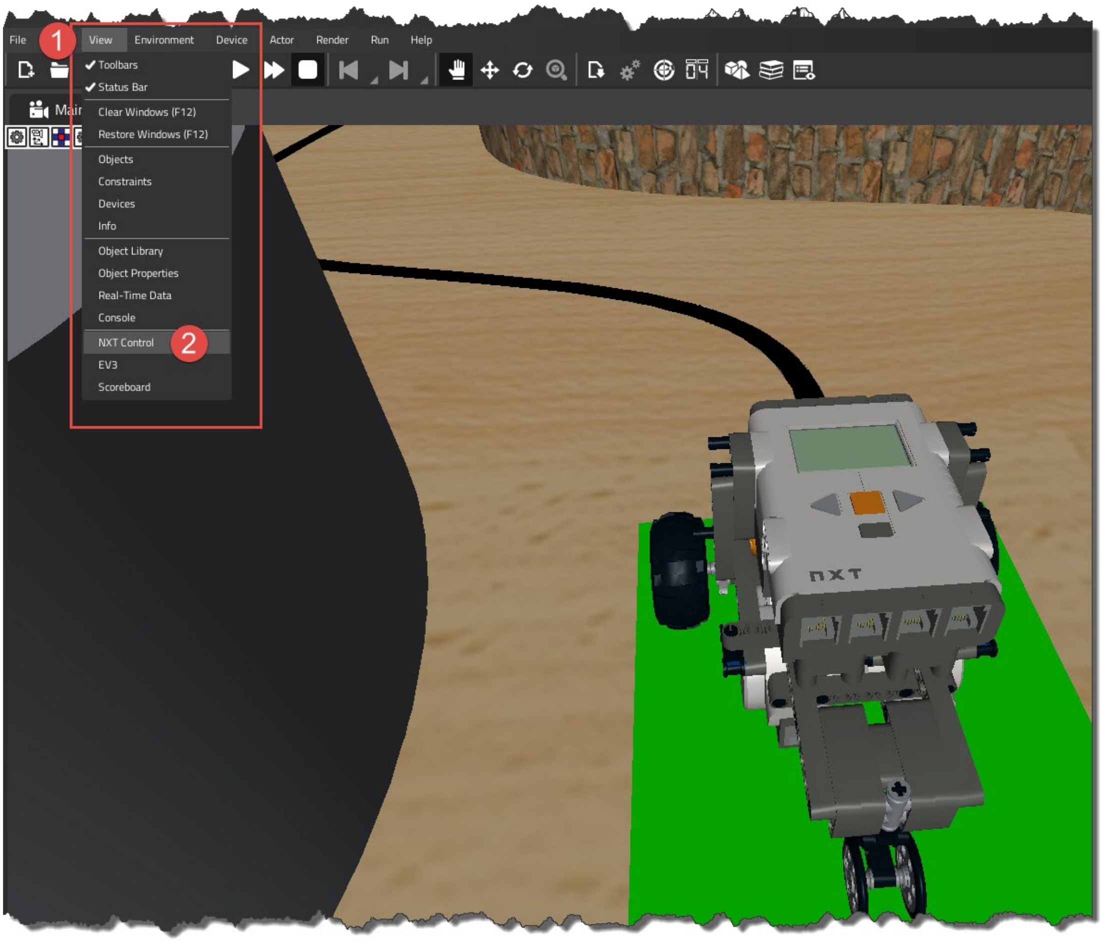
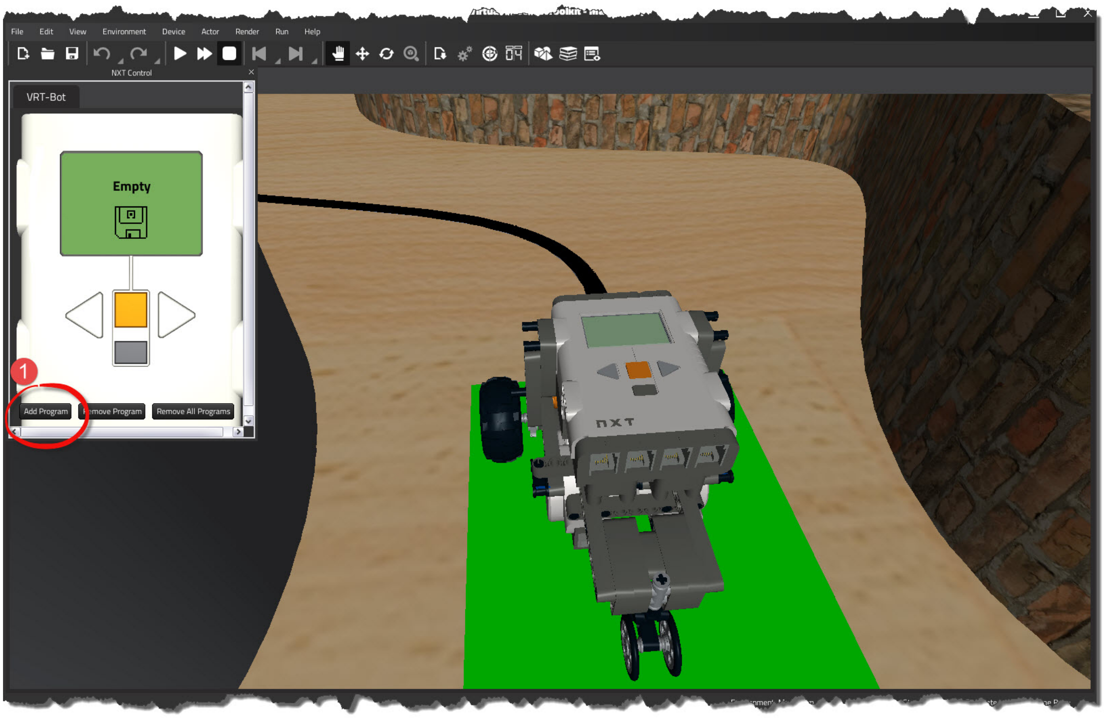


Figure 4.21: Opening the digital NXT Intelligent Brick

6. A digital version of the NXT Intelligent Brick now appears.

7. Click on the "Add Program" button, and find our MoveRobot.rxe file that we saved to the desktop.



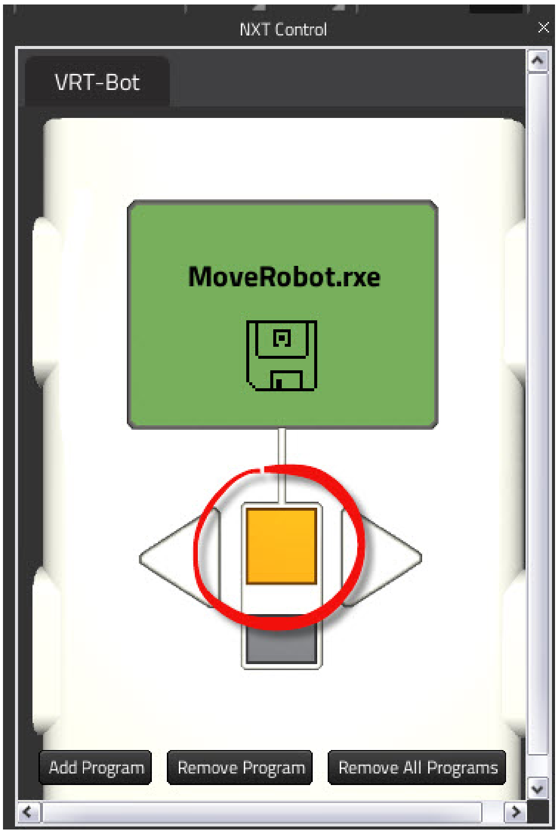


Figure 4.22: Loading program on to brick.

8. On the Intelligent brick, click on the orange centre button.

   If we have done everything right, the robot should move "forward". For troubleshooting see section 4.2.6.

4.2.4 Running our first program in the simulator using EV3

1. Open the EV3 programming environment, and create a new program called MoveEV3. This simple EV3 program will do nothing more than use a single MOVE STEERING block to propel the robot forward one rotation.

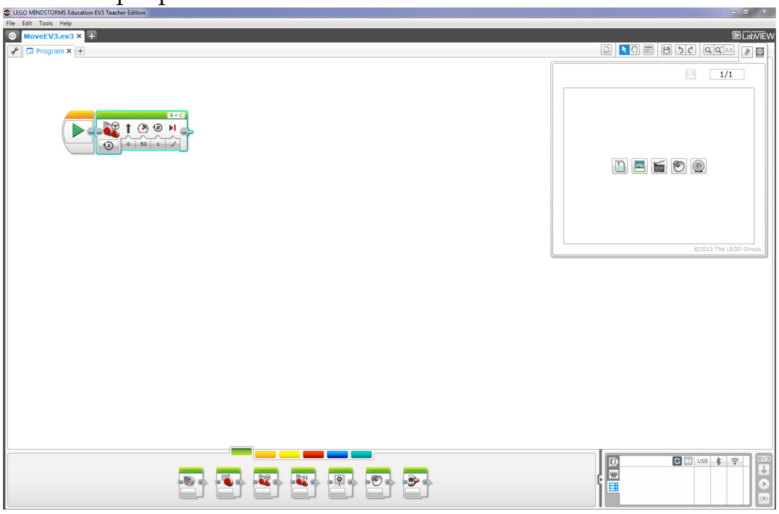


Figure 4.23: Sample EV3 program that uses a move block.

2. In the simulator, click on the View menu and select "EV3" to call up the Intelligent Brick.

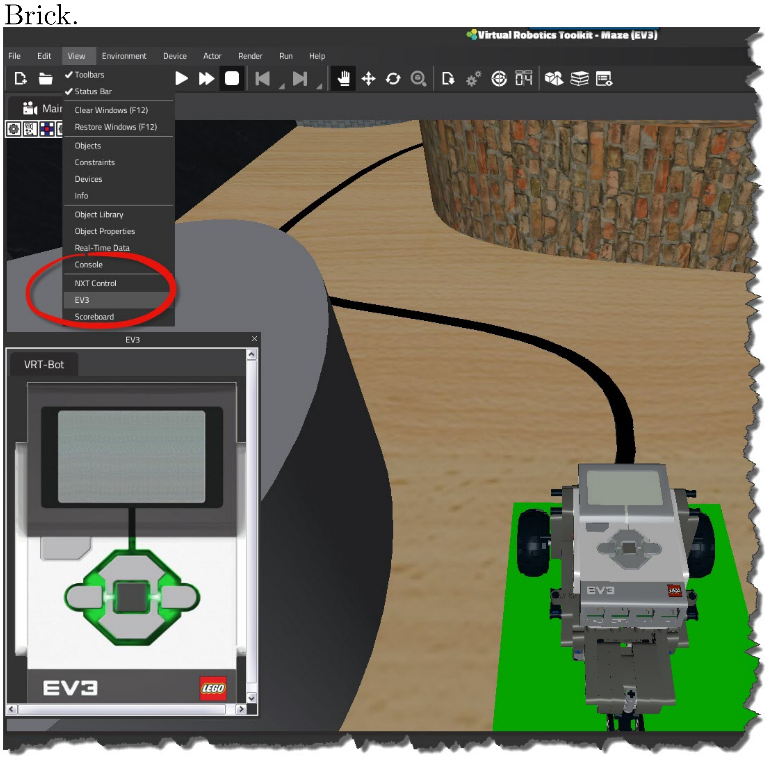


Figure 4.24: Opening the digital EV3 Intelligent Brick.

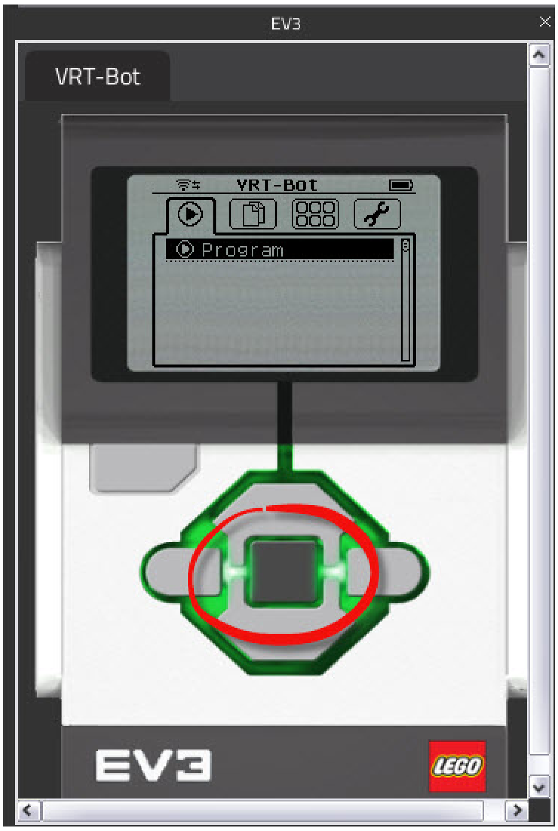


Figure 4.25: Running program on digital EV3 brick.

Turn the digital EV3 brick on by pressing the middle button.

3. In the LEGO EV3 programming environment, note that now under the wifi connections tab, our digital brick appears. Click the download button, to transfer our program to the simulator.

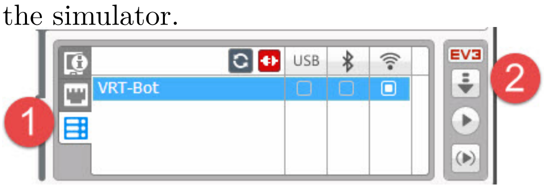


Figure 4.26: Connecting to the simulator from EV3.

4. In the simulator, the digital Intelligent Brick should now have our EV3 program loaded. We can view it by visiting the Bricks' File Navigation tab.

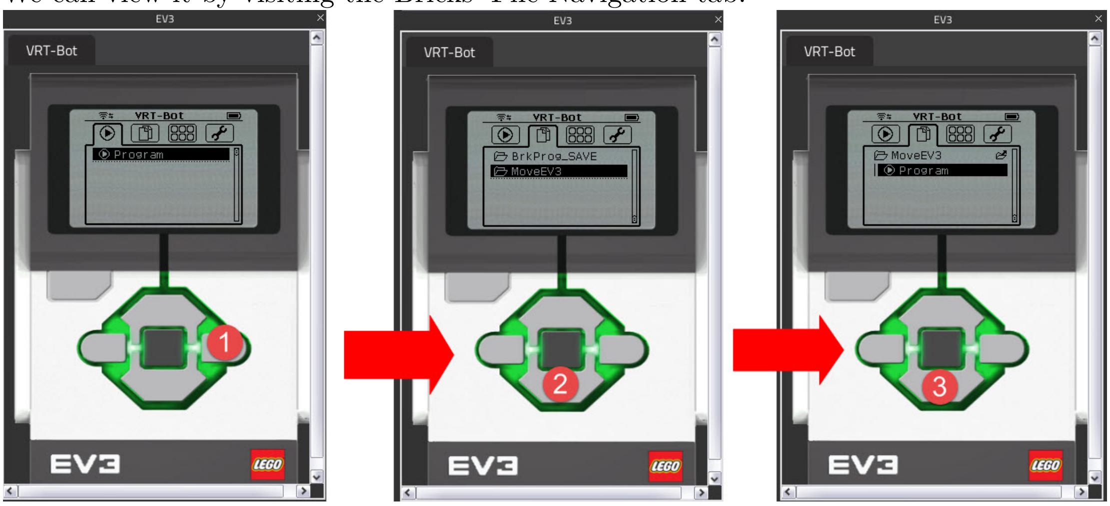


Figure 4.27: Loading a program on to the EV3 brick.

5. Press the Play button to start the simulator, and then press the middle button on the EV3 brick to run our program.

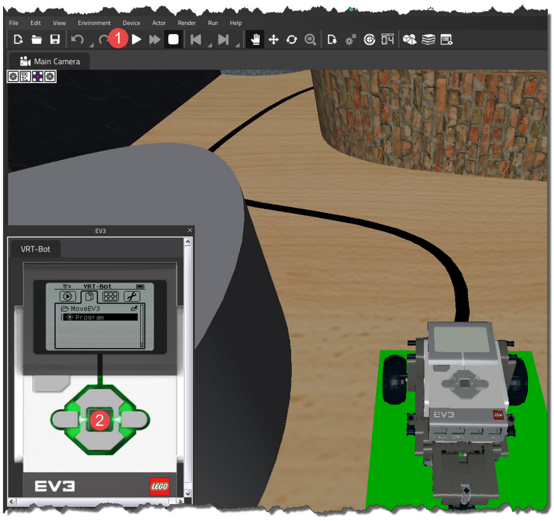


Figure 4.28: Starting the simulator, and then running program off of digital EV3 brick.

6. The robot should move forward one rotation.
7. When finished working with the project, disconnect the brick from the EV3 programming environment.

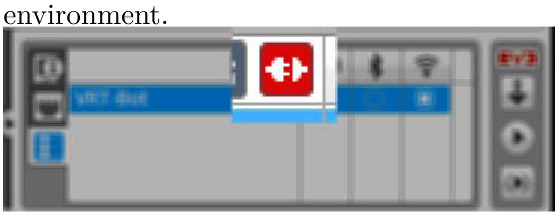
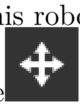


Figure 4.29: Disconnecting simulator from EV3 programming environment.

4.2.5 Working with sensors

In section 4.1.11 we were introduced to the Attachments dialog. We can use many of the sensors that are available in this window to build "smarts" into our robot.
Let's see what attachments are available for this robot.

Select the VRT-Bot by clicking on it with move  mode active and then click on the Attachments toolbar  button.
In this example, we can see that the sample VRT-Bot has two ultrasonic sensors that we can use to help us detect the walls of the maze. Select one of the ultrasonic sensors and add it to the robot, clicking on the "Ok" button will close the **Manage Attachments** window and save the changes we have made to the robot.

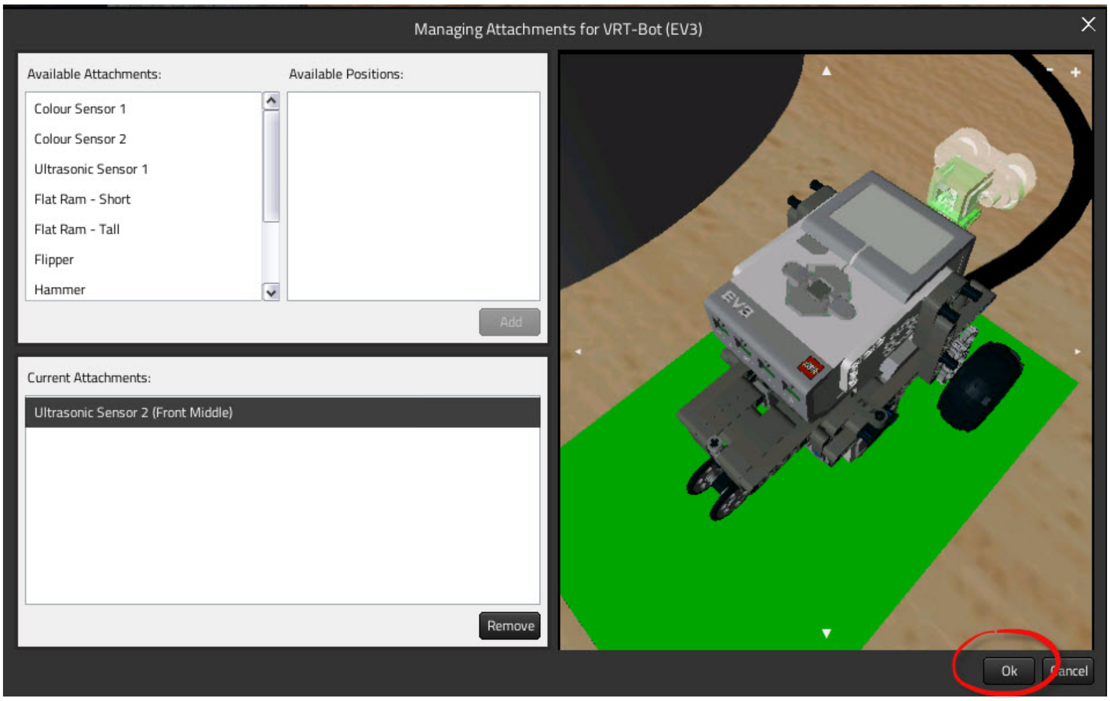
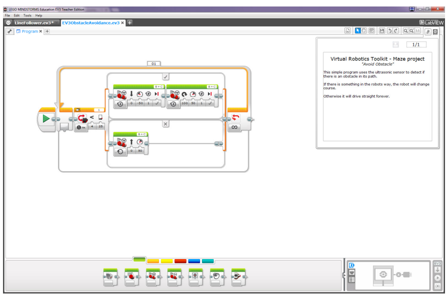


Figure 4.30: Saving added attachments to the sample robot.

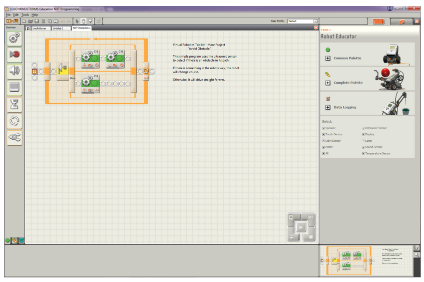
In the LEGO programming environment, write a simple program that uses a SWITCH statement to work with our newly added ultrasonic sensor. For example, if the robot gets with 25 cms of a wall, have the robot stop and turnaround. Otherwise, have the robot drive forward continuously at a set speed of 50.

The EV3 and NXT code are given in links below.

EV3 - "Obstacle Avoidance"



NXT - "Obstacle Avoidance"



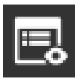
Click screenshots to open

4.2.6 When things go wrong, troubleshooting common programming errors

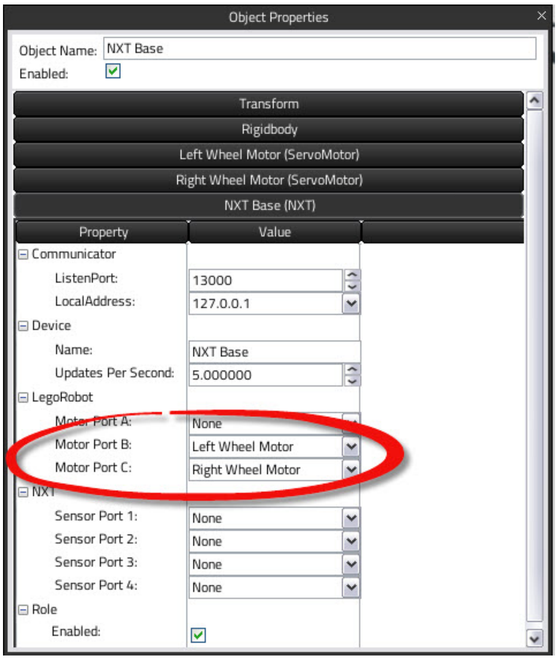
If you happen to run into problems with getting your code to execute in the simulator, it could be because of one of the following reasons:

1. The simulator is not turned on, i.e. the Play  button is not pressed.
2. There is a mismatch between the ports that are being specified in your code, and the ports that your simulated robot is using. To remedy this issue, follow these steps below:

 Select the robot in the simulator

 Open it's Object Properties

(a) Expand the (NXT) or (EV3) label



(b) Observe the ports on the digital brick that are being used.

(c) Open the LEGO programming environment, and check that what is being specified in your move and sensor blocks match with the simulated robot's object properties.

Figure 4.31: NXT configuration panel with matching ports for move block.

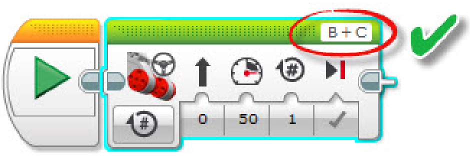
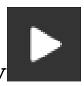


Figure 4.32: Saving added attachments to the sample robot.

4.2.7 The Real-Time Data display

One of the simulator's more interesting features, is the ability to provide real-time data from the simulated environment.

To view this information, start the simulator by pressing the **Play**  button , and then click on the **Real-Time Data**  toolbar button.

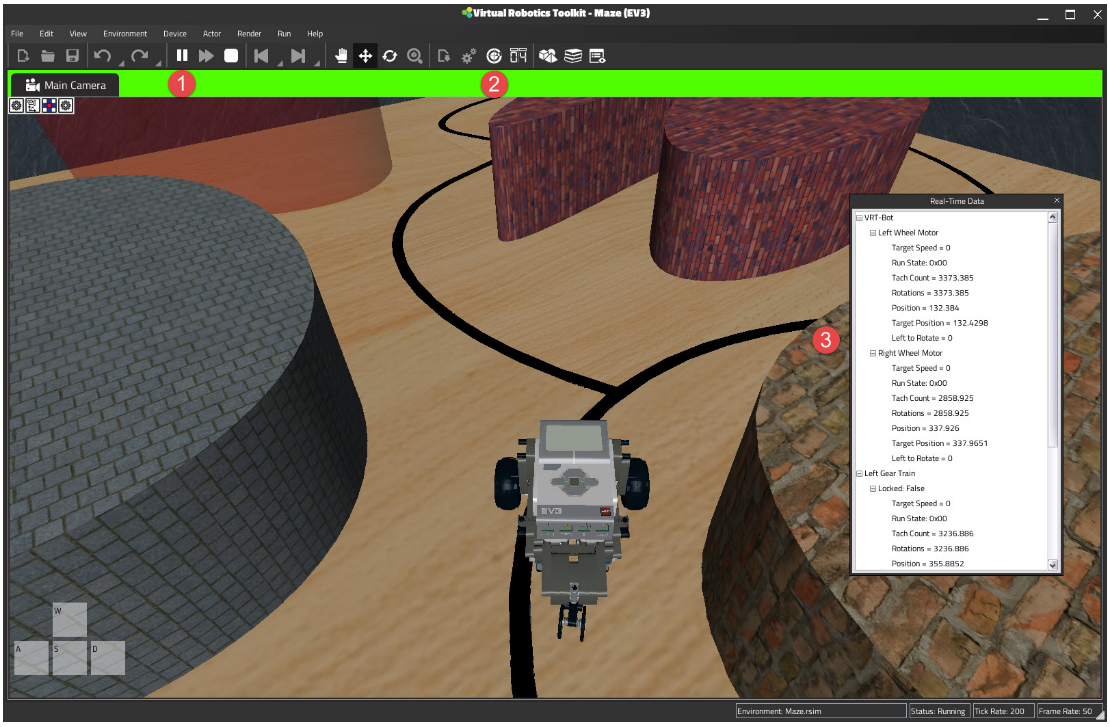
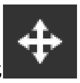


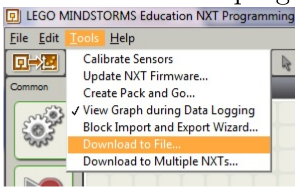
Figure 4.33: Real-Time Data.

4.2.8 Finding our way through the Maze

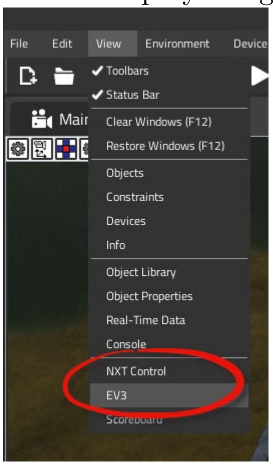
This section will conclude by discussing how to work with sensors, and demonstrate how we can use the information they provide to navigate the Maze.
Starting at the green square, the objective is to use the colour sensor to follow the black line through the maze and have the robot stop at the red square.
Try this now on your own, if you get stuck, working EV3™ and NXT™ project code is supplied below.
Helpful Hints:

- Select the robot by using the **Move Object**  button.
- Open the **Attachments**  dialog, and add a colour sensor. When finished, click "Ok" to close the Attachments Manager and to save the changes you have made to the robot.
- Next, **start the simulator** with the **play**  button and open the **Real-Time Data**  window

- Begin driving the robot. What values does the "Real-Time Data" window report for the colour sensor? How might we use this information in our LEGO® MINDSTORMS® program.



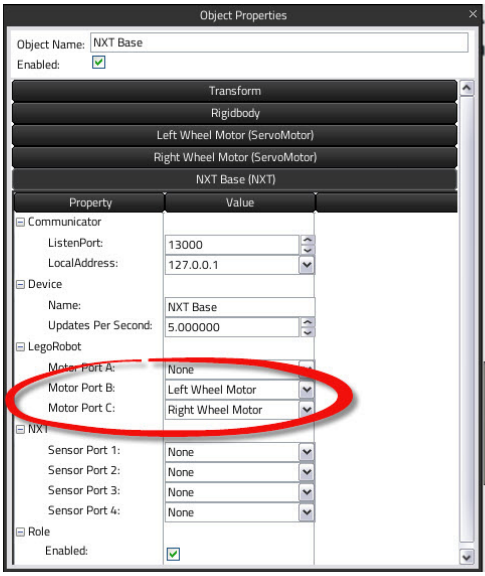
- Write your program.
- If using NXT-G,click on the "Tools" menu to save your compiled program to the desktop.
- In the simulator, load your program to the Intelligent Brick. The digital brick can be called up by using the **View** menu.

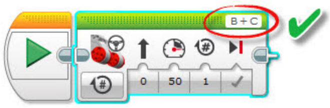


- If using EV3, turn the brick on and connect to it through the EV3 programming environment. The digital brick should show under your Wifi connections tab.

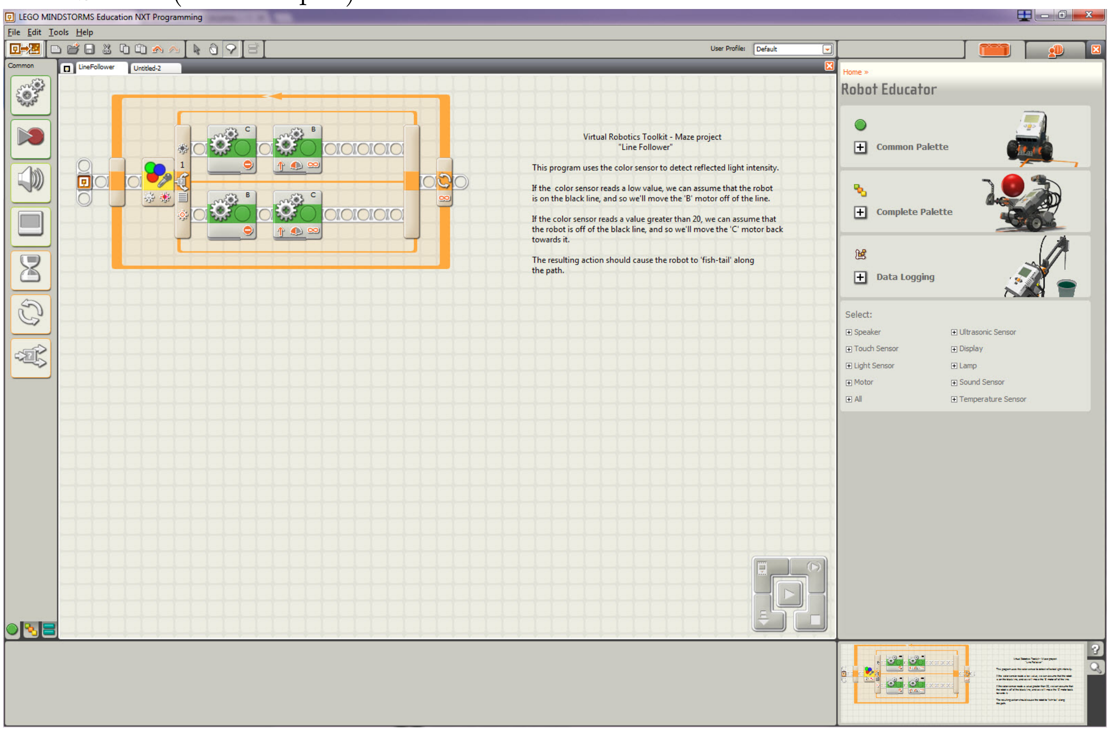
- Press **Play**  to start the simulator.
- Run the program on the brick as shown here in Figure 4.25.

- Does it work? If not, check your port settings in the **Object Properties**  window, and compare against what is specified in the MINDSTORMS® programming environment.

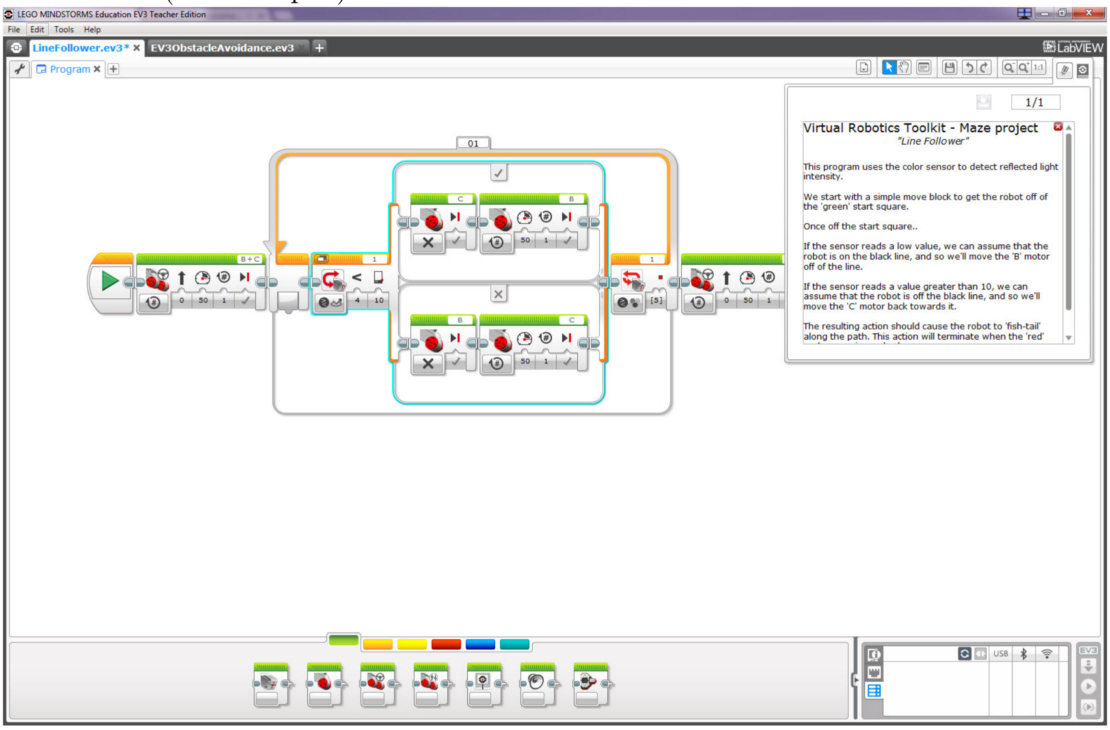




NXT Source (click to open)



EV3 Source (click to open)



4.2.9 Summary

In this section we used the Maze project to learn how attachments work. We also saw that by clicking on the Real-Time Data button, we could literally access a treasure trove of data about the robot and its environment. This information can later be used as a basis for writing a MINDSTORMS® program to help navigate the robot around the Maze.

So far, our focus has been on controlling the sample robot that is provided. However, the real magic of LEGO® robotics comes from the ease with which we can design our own creations.

In the next section, we'll outline the process for importing a custom model created using LEGO® Digital Designer™.

4.3 Building

4.3.1 Project overview

The purpose of the Cleanup challenge is to clear as many blocks as possible in the fastest time without falling off of the mat.

The sections that follow will introduce LEGO Digital Designer as a modelling tool, and will then detail the process for importing our own robot into the simulator.

We will need to design a robot that has a suitable apparatus for clearing the field, as well as sensors for detecting our position in Model Space. To do this, we must first build the model using LEGO® Digital Designer™.

4.3.2 Introducing LEGO® Digital Designer™

As was first mentioned in Chapter 2, LEGO® Digital Designer™ (LDD), can be downloaded by clicking on the link below:

`http://www.lego.com/en-us/MINDSTORMS/downloads`

Using LDD is a straightforward affair, as in the opinion of the author it is quite an intuitive tool to use. The "Help" documentation that exists within LDD will give you a very good foundation for getting started with building models of your own. The best part of using LDD is that in addition to being able to create exportable simulator models, it can also be used to generate a Bill of Materials and printable building instructions for creating the physical robot.

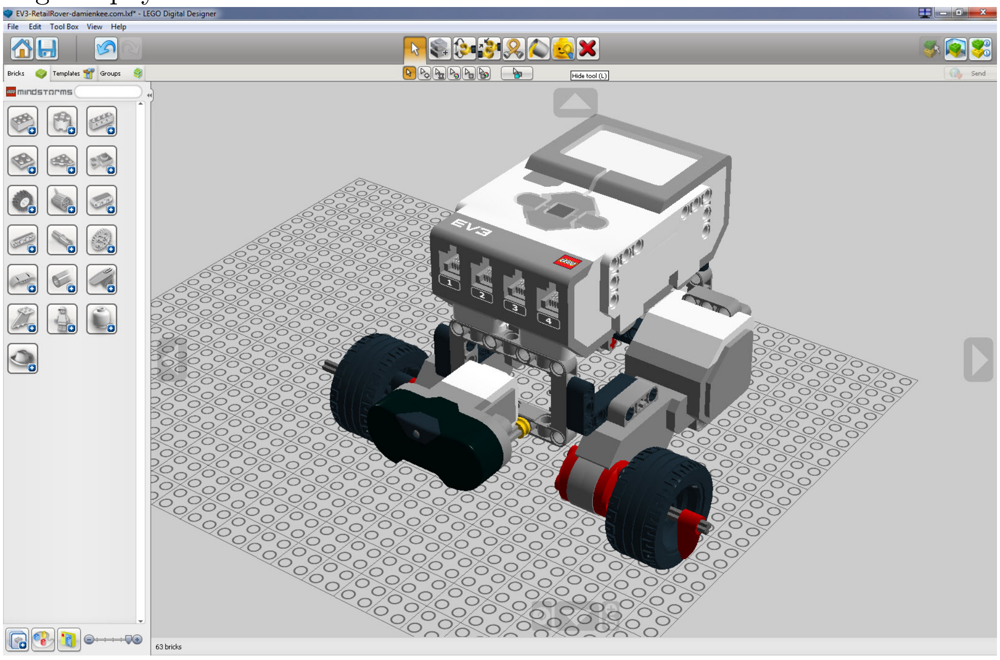


Figure 4.34: LEGO Digital Designer.

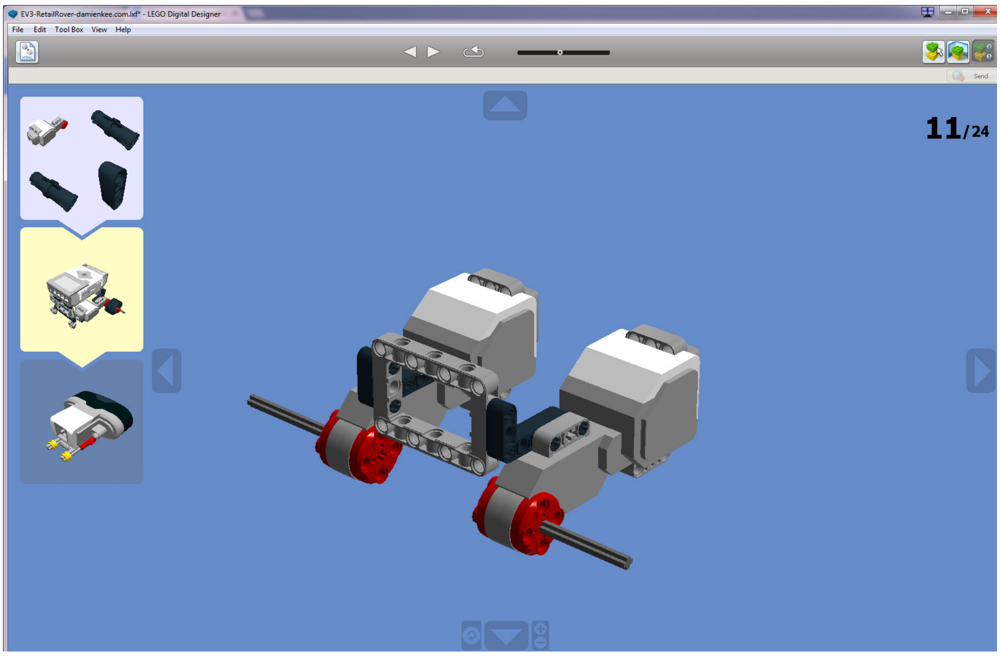


Figure 4.35: Create printable instructions using LDD's building guide mode.

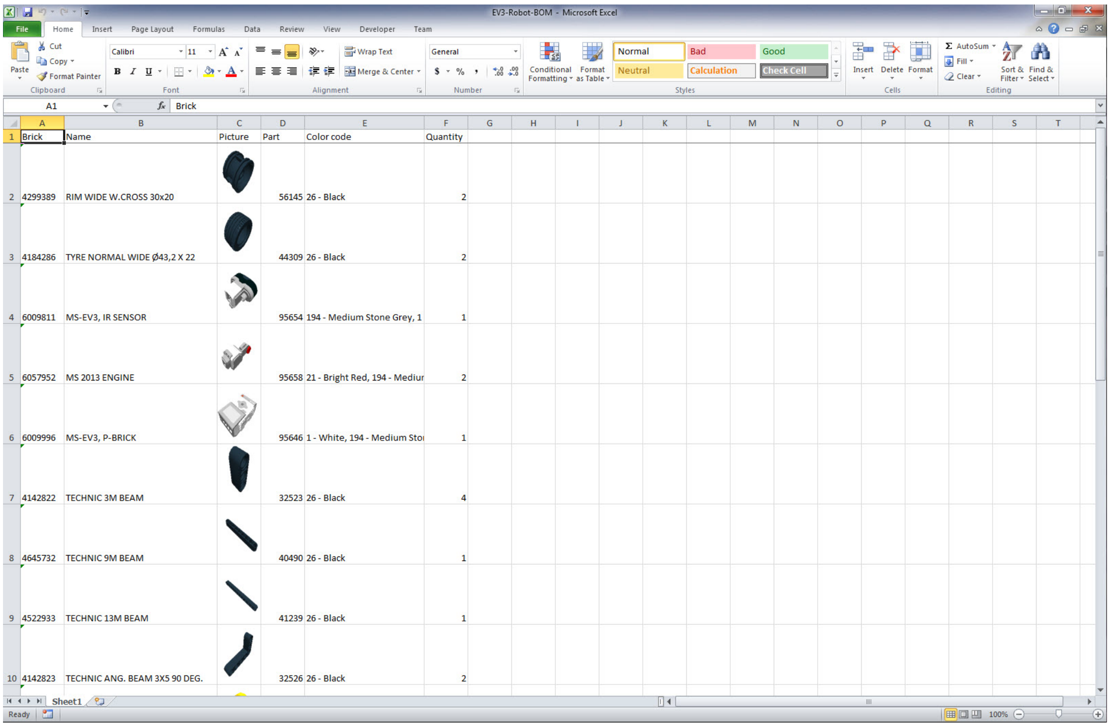


Figure 4.36: Bill of Materials created from LEGO Digital Designer.

4.3.3 Alternate design tools

The simulator can also work with models that are created in MLCad, as well as any number of other applications which are capable of exporting files to LDraw format (*.ldr).

4.3.4 What is LDraw?

LDraw is a popular CAD format for sharing 3D LEGO models. The simulator is capable of parsing an imported (*.ldr) file, and then assigning physical attributes to it such as weight, centre of mass, and collision data to each model element.

If you haven't already done so, you should consider downloading and installing the LDraw "AllinOne" installer, as it will greatly expand the number of bricks that the simulator can work with.

`http://www.ldraw.org/help/gettingstarted.html`

4.3.5 Importing into the sim

The sequence of steps for importing an LDraw model into the simulator will now be outlined.

1. In the Simulator, click on the Help menu and select Patch LEGO® Digital Designer™.
2. In LEGO Digital Designer, click on the "File" menu and choose "Export Model".
3. When prompted, choose LDraw (*.ldr) from the list of exportable file formats and a location to save your file to.
4. Open the simulator, and select the Cleanup challenge.
5. Click on the **Move Object** button and select the robot. With the robot selected,

press the delete key to remove it from Model Space.

6. We should now be looking at a project that has no robots in it.
7. To import our custom robot into the simulator,click on the **Environment** menu and choose "Import Model..."
8. This will launch the Simulator's LDraw import wizard.
9. Once imported, the simulator will analyze the model to determine which components are rigidly connected (i.e. connector pegs, and elements that make up the chassis), and which components are free moving (i.e. the wheels on the robot).
10. Next, the import wizard will identify special components such as motors and sensors. The user will be asked to assign these special components to ports on the Intelligent Brick so that they may be later controlled by the LEGO® MINDSTORMS® programming environment.
11. With motors and sensors mapped to their respective ports, the user will then be asked to assign a set of keyboard controls for driving the robot. Being able to take the robot for a test drive in the simulator to see how it handles can be an especially invaluable tool for competitive robotics teams before they invest considerable time and effort into programming the robot to complete challenges.
12. The model will then be finalized, and the user will be prompted to click "Ok" to close the wizard.
13. The robot now appears in the center of the project file, pressing the **Play** button to start the simulator allows us to drive our creation using the keyboard keys that we assigned.
14. As a curiosity, select the newly imported robot using the Move Object tool. Click on the Render menu and make sure "Physics" is checked. Note that now the imported model lights up like a christmas tree! That's because the model now contains intelligence, it's no longer just a "fancy" picture.

4.3.6 Using your robot to complete the Clean-up challenge

Try now to design a cleanup robot of your own, and import it into the simulator by using the LDraw import wizard.

Once imported, drive your robot using the keyboard keys or develop a simple program using move blocks in the MINDSTORMS® programming environment.

If creating autonomous behaviours using the LEGO® programming environment, be sure to include a colour sensor in your design so that you can use a switch block to detect the edge of the mat.

4.3.7 Summary

If you have made it this far, you're now in a very good place. With the basics of simulator control, programming, and design covered. We're ready for our capstone project, the Apartment. Here we'll program a robotic vacuum cleaner to tidy a room.

4.4 Putting it all together

In this chapter, we saw how we could use the simulator to design, program and test our robot by pairing the Virtual Robotics Toolkit with a number of freely distributed LEGO tools.

4.4.1 Project overview

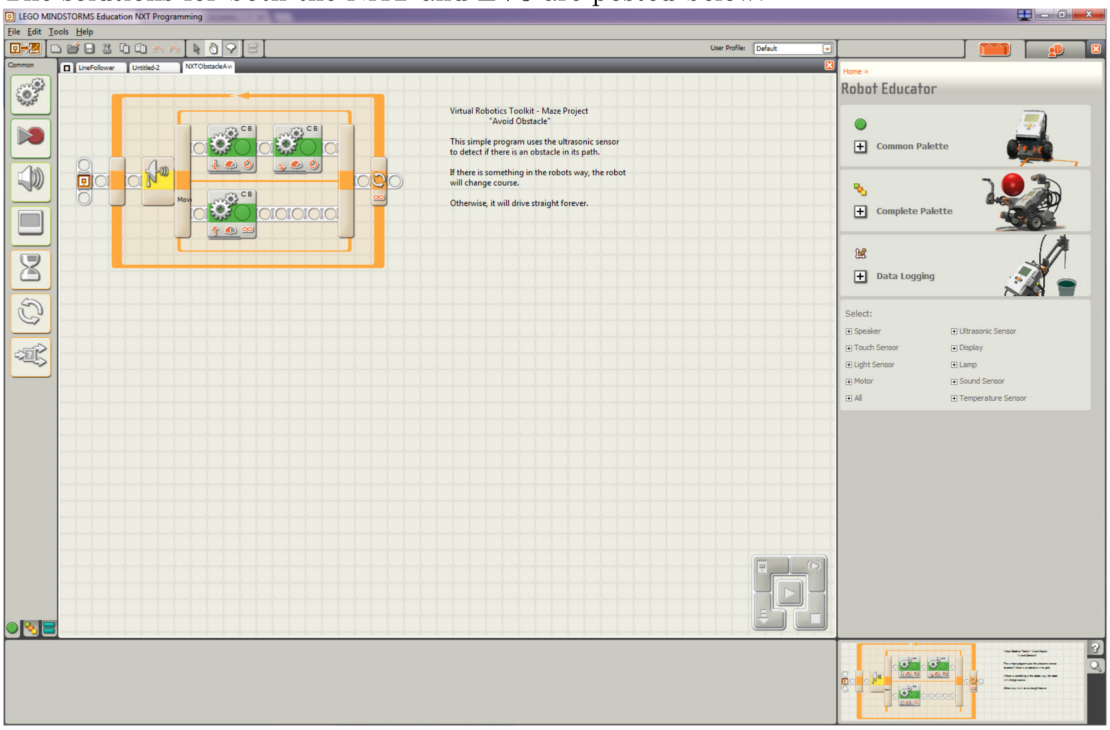
The goal of the Apartment challenge is to use our understanding of the simulator's attachments dialog, as well as its basic controls, and to create a MINDSTORMS® program to vacuum all of the blocks that litter this room.

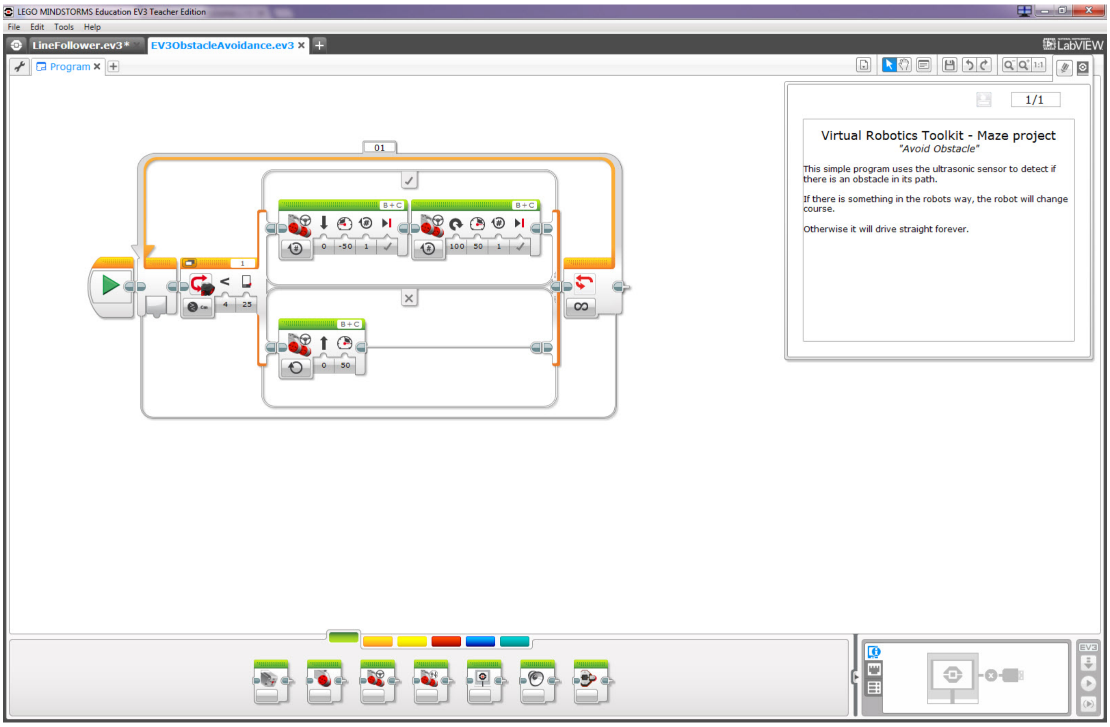
4.4.2 Helpful hints

- How might this robot detect objects in its path?
- What should happen if an object gets in the way of our robot vacuum cleaner?
- Are there any sensors that we can use from the simulator's attachments dialog?
- Go ahead, and write a simple MINDSTORMS® program that uses the attached sensor's SWITCH block to make a decision:
  - If an object blocks the robots path, backup and turn around.
  - Otherwise, drive forward continuously.
- Make sure that the ports for the motors and sensors on the simulated robot match with what is being specified in the LEGO programming environment. To check these settings in the simulator:
  - Select the robot by using the **Move Object** toolbar button
  - Click on the **Object Properties** toolbar button.
  - In the properties window that appears, expand the (EV3) or (NXT) label and check on the port mapping.

4.4.3 Solution

The solutions for both the NXT and EV3 are posted below.




LEGO MINDSTORMS Education EV3 Teacher Edition
File Edit Tools Help
LineFollower.ev3*
EV3ObstacleAvoidance.ev3
LabVIEW
Program
1/1
Virtual Robotics Toolkit - Maze project
"Avoid Obstacle"
This simple program uses the ultrasonic sensor to detect if there is an obstacle in its path.
If there is something in the robots way, the robot will change course.
Otherwise it will drive straight forever.

5. Challenges

5.1 Multiplayer Sweep

5.1.1 Overview

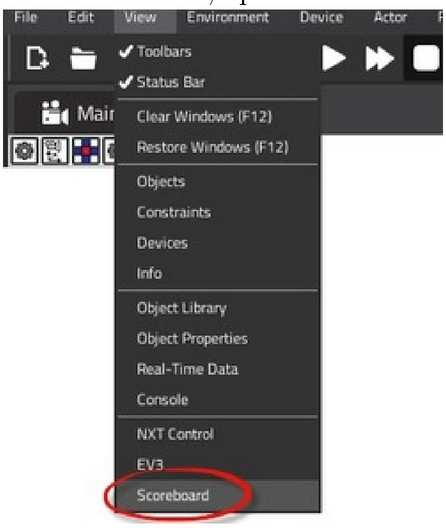
This is a multi-robot version of the Clean-up challenge featured on Tab 1 (Sim Basics)

5.1.2 Objectives

- Score as many points as you can by pushing the blocks into your colored bin
- Push your opponent off the edge to eliminate them for 5 seconds
- The bigger blocks are worth more points:
  - Red Block (x1) = 250 Points
  - Blue Block (x1) = 200 Points
  - Yellow Block (x1) = 150 Points
  - Green Block (x1) = 100 Points
  - Pillars (x9) = 50 Points

5.1.3 Hints

In the simulator, open scoreboard by clicking on **View** menu → **Scoreboard**.



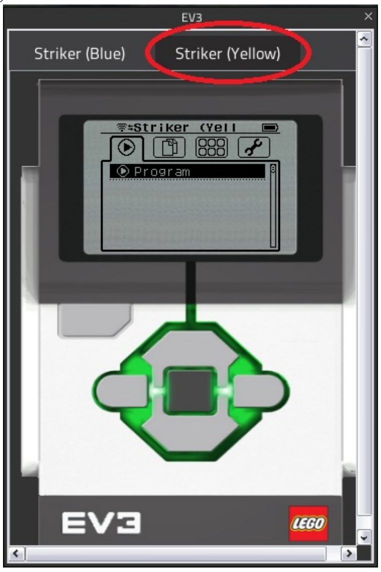
There are multiple strategies to this challenge, focus on:

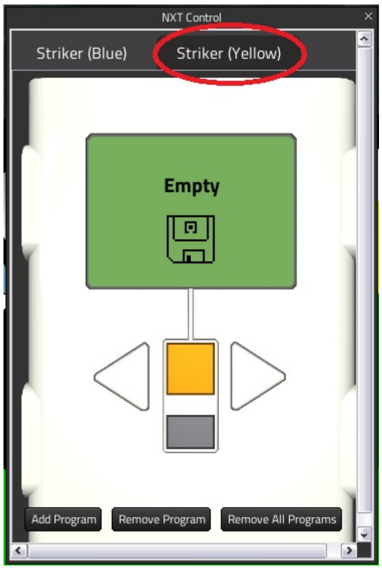
- Pushing the opposing robot off will allow you more time to collect blocks before they respawn
- Getting the big blocks first
- Get the small pillars
- Be the fastest to collect any points without falling off

There are many methods to get the highest score, here are some suggestions:

- Bounce around while avoiding the edge
- Find the opponent and push them off
- Find the bricks based on color sensor
- Find the bricks based off of the UltraSonic Sensor

To load a program to control the opposing robot, click on the tabs within the Brick display.





5.1.4 Pseudo Code

1. Start Program
2. Commence infinite loop
3. **IF** color sensor detects white boundary
   (a) BACK-UP
   (b) TURN AROUND
4. **ELSE** drive forward at ramming speed!
5. Return to line 3, loop forever

5.2 Soccer

5.2.1 Overview

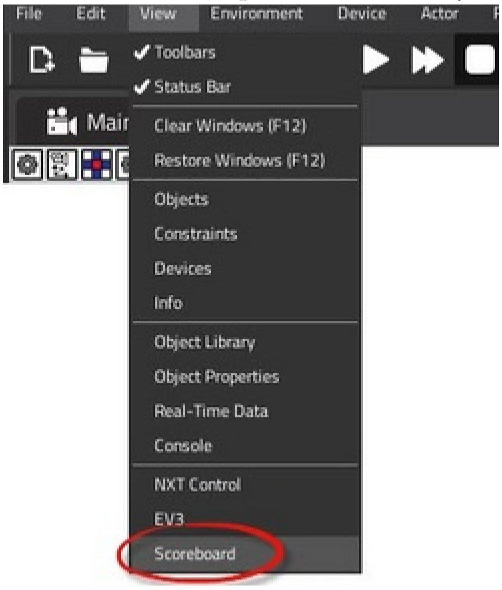
This is a virtual version of the World Robot Olympiad GEN II soccer field.

5.2.2 Objectives

- Score as many goals against your opponent as you can!

5.2.3 Hints

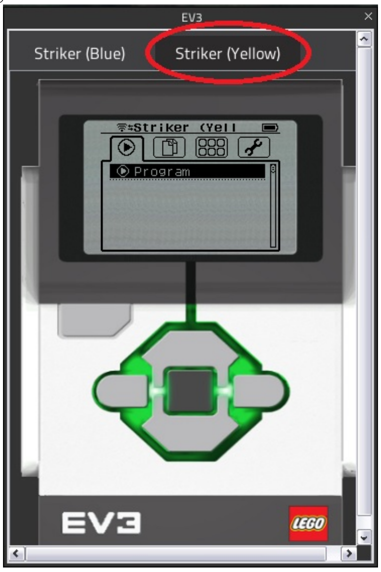
In the simulator, open scoreboard by clicking on **View** menu → **Scoreboard**.

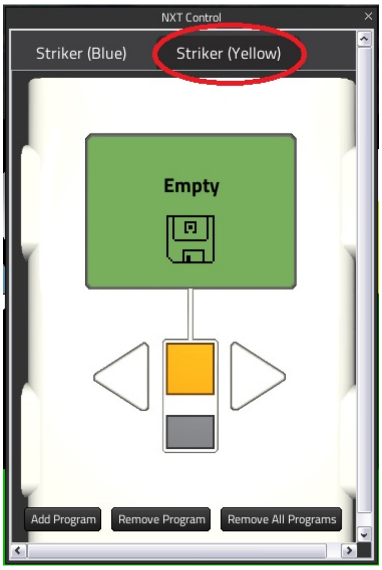


Use the IR Seeker sensor and Compass sensor to find the ball, and locate the opposing net.

NOTE: You require the programming blocks from HighTechnic® in order to use the IR Seeker and Compass Sensors. These programming blocks can be downloaded from this link: `https://www.hitechnic.com/downloads`

To load a program to control the opposing robot, click on the tabs within the Brick display.





5.2.4 Pseudo Code

1. Start Program
2. Commence infinite loop
3. **IF** robot has the ball (Use color sensor to determine if the robot has the ball)
   (a) Use compass sensor to point at the net
   (b) Shoot the ball
4. **ELSE**
   (a) Use the IR Seeker sensor to point at the ball
   (b) Drive forward
5. Return to line 3, loop forever

5.2.5 EV3 Sample Code:

Click **HERE** to open file.

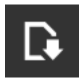
5.3 Sumo

5.3.1 Overview

This is a robotics battle that is 400 years in the making! The traditional Japanese sport of Sumo wrestling flourished in the 1600's throughout feudal Japan. Now, design and program a robot to push your opponent out of the ring.

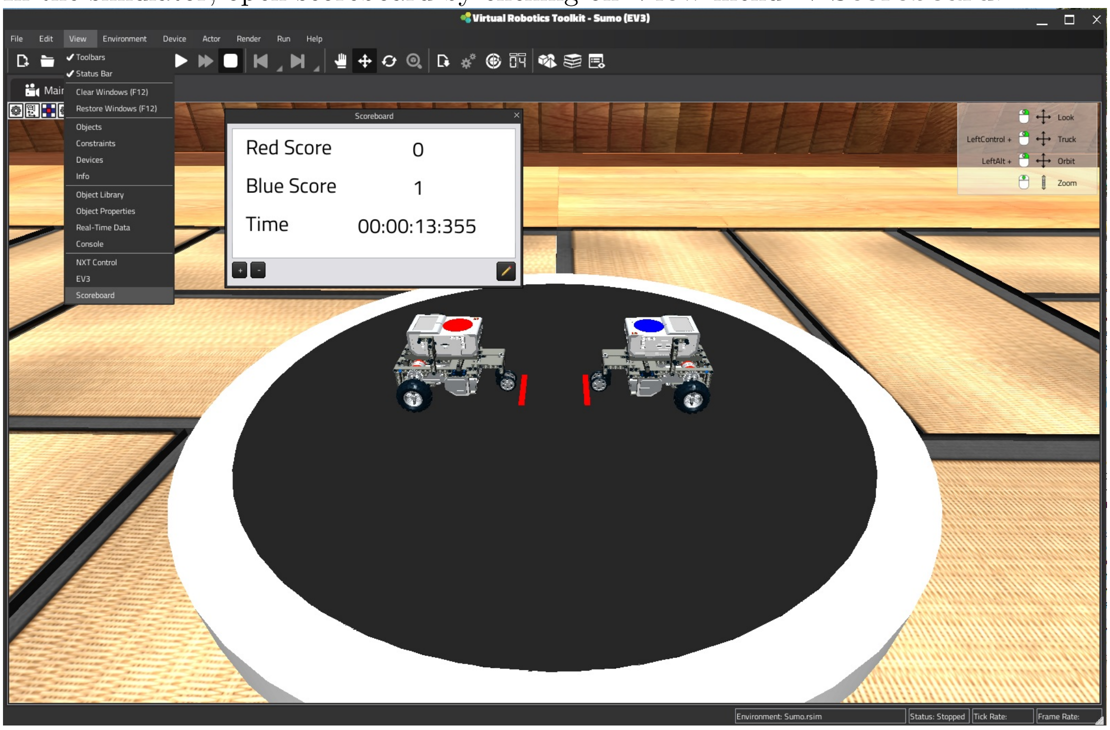
5.3.2 Rules

- Robots must face away from each other with their back wheels placed at the red hash marks in the center of the ring.
- Any assortment of sensors or attachments may be used to create an advantage.

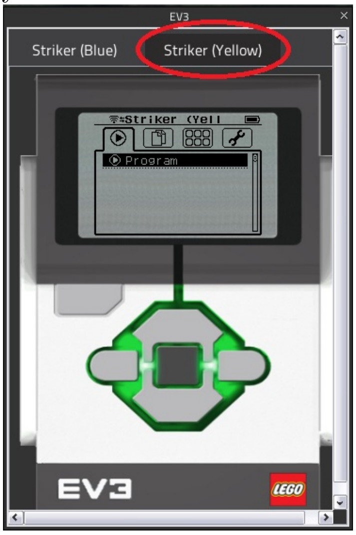
- Alternatively, use the LDraw wizard to design your own robot. 
- A point is scored when one robot pushes the other outside of the ring and onto the floor.
- The robot with the highest score after 3 minutes of battle is the winner.

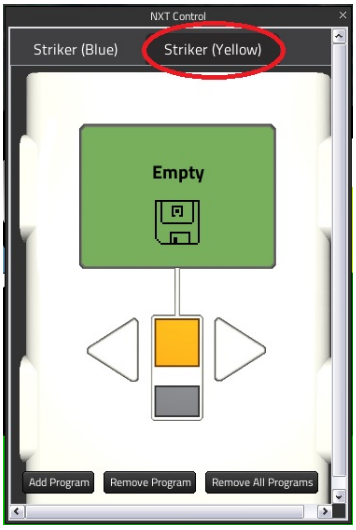
5.3.3 Hints

- In the simulator, open scoreboard by clicking on **View** menu → **Scoreboard**.



- To load a program to control the opposing robot, click on the tabs within the Brick display.





5.3.4 Pseudo Code

1. OUTER LOOP
2. INNER LOOP #1 //find the other robot, loop forever until opponent is detected.
   (a) IF edge of ring is detected
      i. BACKUP
      ii. TURNAROUND
   (b) ELSE
      i. Drive forwards
3. INNER LOOP #2 //charge at opponent, stop when edge of ring is detected.
   (a) CHARGE FORWARDS
4. BACK-UP to avoid leave ring.

5.3.5 NXT Sample Code:

Click **HERE** to open file.

6. Advanced features and technical stuff

In this final chapter, we?ll discuss some of the more advanced settings that are available to us. While, it is highly recommended that you refrain from changing these default settings, if you do happen to have curious fingers, this is where you can adjust many of the underlying assumptions that are being made by the software's physics engine.

6.1 Preferences menu

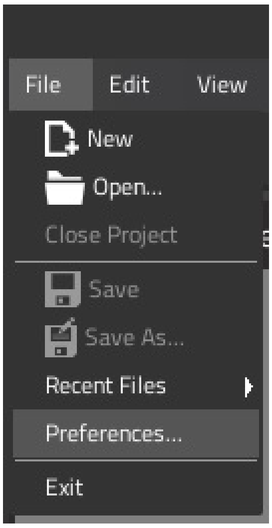


Figure 6.1: Starting with the **Preferences** dialog, it is here that we can change settings that govern the display quality of the simulation. This window is divided into four tabs:

1. General tab
   - **General Settings**: Allow us to set an upper bounds on the number of undo commands that the simulator can keep track of.
   - **Display**: Allows the user the adjust the overall display quality of the simulator. If running the software on an older machine, consider changing this setting to "fast".
   - **Viewport**: These settings control the near and far plane of the viewing frustum, the region of space in the modelled world that is visible on screen.

Anything closer to the eye than the near clipping distance isn't displayed (it's too close), and anything further away from the eye than the far clipping distance isn't displayed either (it's too far away). The Background Colour allows us to set a default background in lieu of a skybox.

2. Input tab
   - The settings in this tab correspond to the keyboard / mouse mappings of various camera controls. These can be adjusted to suit your own preferences. Alternatively, to restore these settings click on the button in the bottom right hand corner of the window.
3. Network tab
   - Allows third party software to communicate with the sim using the specified IP address and port.
4. Other tab
   - **Unit System**: allows the user to toggle between using imperial and metric systems of measurement.
   - **Angles**: allows the user to toggle between using degrees and radians for measuring angles.
   - **Coordinate Display**: will toggle the display of the X,Y,Z, coordinates on the bottom left hand side of the screen.
   - **Controls Display**: will toggle the on screen keyboard and camera controls which are overlaid on top of Model Space.
5. **LDraw tab**: specifies the location of the LDraw parts inventory which is used by the simulators' LDraw Import Wizard.

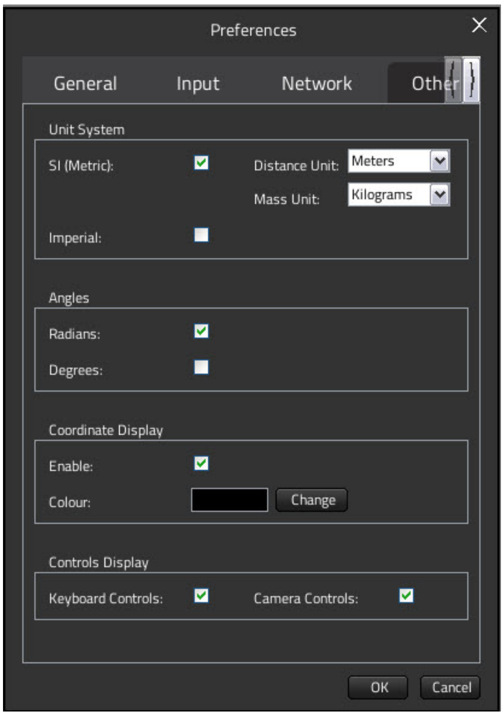


Figure 6.2: Preferences menu with tabs

6.2 Environmental settings

The simulator's environmental settings allow us to make adjustments to the actors, lighting, physics, and triggers that make up a project.

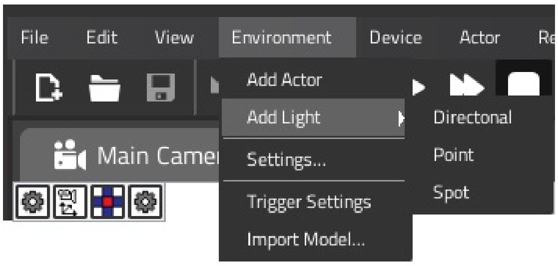


Figure 6.3: The Environment menu

Actors represent the "things" or objects that the simulator must keep track of, and can contain one or more components. For example, a robot (actor) may have various motors or sensors (components) assigned to it. An actor's components determine the role that it will play in the simulation. An actor with no components is often used as an organizational construct for the purposes of grouping similar objects together. Actors and components are described in greater detail in section 6.4.

The **Add Actor** option in this menu will create a new actor without any components in the **Objects** window.

Lighting allows you to adjust the illumination of Model Space by adding or adjusting light sources. The different kinds of light sources that are available include:

- **Directional lights**: these types of lights are placed infinitely far away and affect everything in the scene. A directional light is the best way of emulating the sun.
- **Point lights**: are lights that shine equally in all directions, much like a light bulb.
- **Spot lights**: are lights that illuminate everything within a directed cone. Only objects within this region are affected by the light.

The **Add Light** option in this menu will create a new light of the specified type to Model Space.

To change the physical properties of the environment, we must click on the **Settings** menu item. The Physics tab governs the physical behaviour of the environment, whereas, the Environment tab controls its appearance.

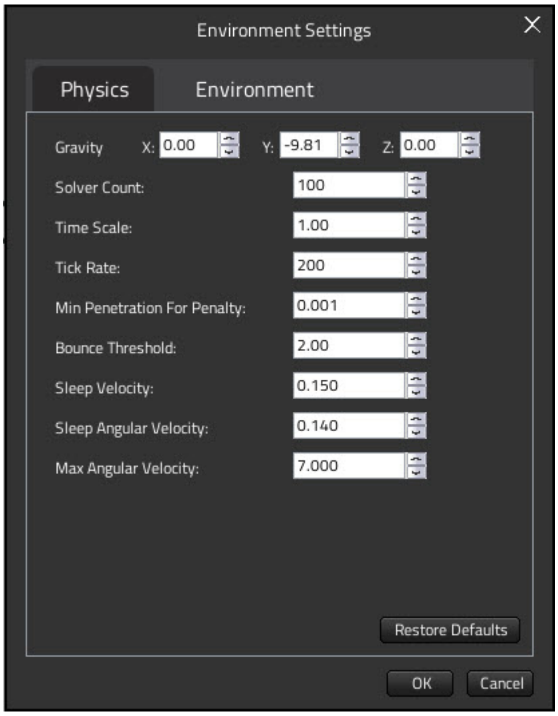


Figure 6.4: Environment Settings Physics tab

1. Physics tab:
    - Gravity: controls the direction and strength of the gravitational force experienced by objects.
    - Solver Count: determines how accurately interactions between objects are resolved. A high solver count prevents the oscillating of objects and other erratic behaviours.
    - Time Scale: allows you to speed up or slow down simulation time. For example, a value of 2 would make time go by twice as quickly. Alternatively, a value less than 1 would have the effect of running everything in slow motion.
    - Tick Rate: refers to the number of physics iterations per second. A higher value will result in a more realistic simulation, but will require additional performance.
    - Min Penetration For Penalty: defines how far objects are allowed to "clip" or penetrate into each other when they come into contact.
    - Bounce Threshold: is a threshold that controls how objects are allowed to bounce off of each other. Two colliding objects with a relative velocity below this value will not bounce.
    - Sleep Velocity: determines a threshold velocity below which objects will be considered to be "sleeping" or stationary. Objects that are "sleeping" will keep their state until acted upon. This is done as means of boosting performance

by avoiding velocity calculations on objects that are not moving.

- Sleep Angular Velocity: determines a threshold angular velocity below which objects will be considered to be "sleeping" or stationary. Objects that are "sleeping" will keep their state until acted upon. This is done as means of boosting performance by avoiding velocity calculations on objects that are not moving.
- Max Angular Velocity: determines the maximum angular velocity at which objects are allowed to rotate.
- Restore Defaults button: will reset all parameters to their original values.

2. Environment tab allows us to change the cosmetic appearance of the environment. For example, we can change the project's skybox or enable fog.

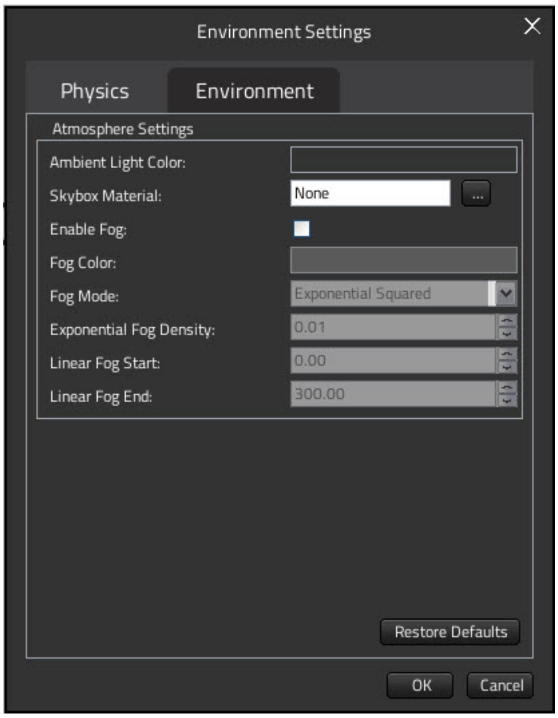


Figure 6.5: Environment Settings Environment tab

Trigger settings menu option will bring up a list of all the interactive elements in the project. Regions which are marked as triggers will appear here. These are regions of special importance, for example a scoring region around the inside of a soccer net, or the floor outside of a sumo ring. Below each region is a list of events that can fire when objects enter or exit the region. For example, when a soccer ball enters a scoring region, increase the teams' score by one.

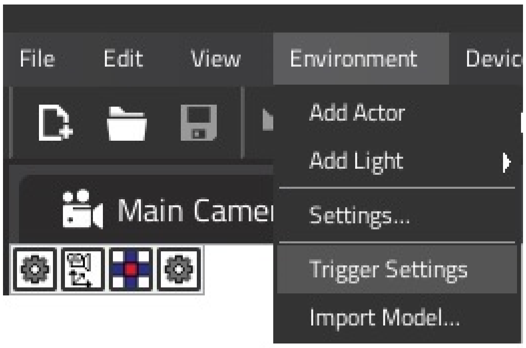


Figure 6.6: Locating the Trigger Settings

6.3 Device Controls

The Device Controls dialog is where you can edit the keyboard/mouse/joystick controls for your robot.

Figure 6.7: Locating the Device Controls

All editable controls are shown in the left tree in the "All Devices" column.

- **Adding Controls**: To start editing controls, select a robot from the "All Devices" column and assign keyboard keys (or game controller buttons) to each "Input" field. In the "Value" column, set the desired speed of each motor by entering a positive number to move clockwise, or a negative number to move counter-clockwise.
- **Detailed View**: Clicking the checkbox for "Detailed View" will add additional columns to the window that can be further used to map motor movement to specific keyboard / mouse events.. Such as when a button "Is Down" or "On Release".
- **Default controls**: The default control keys can be easily mapped to a motor by selecting that motor from the "All Devices" list and pressing the "Add Default Controls" button. This will map the specified motor to the W,A,S,D keyboard keys. W,A,S,D is a commonly used keyboard convention in most PC games. The ability to assign default controls is only available in the "Detailed View".
- **Saving controls**: To save your mouse / keyboard controls, click on the Save button in the bottom right hand corner of the window. Clicking on the Cancel button instead, will close the Device Controls window without saving any of the changes made.

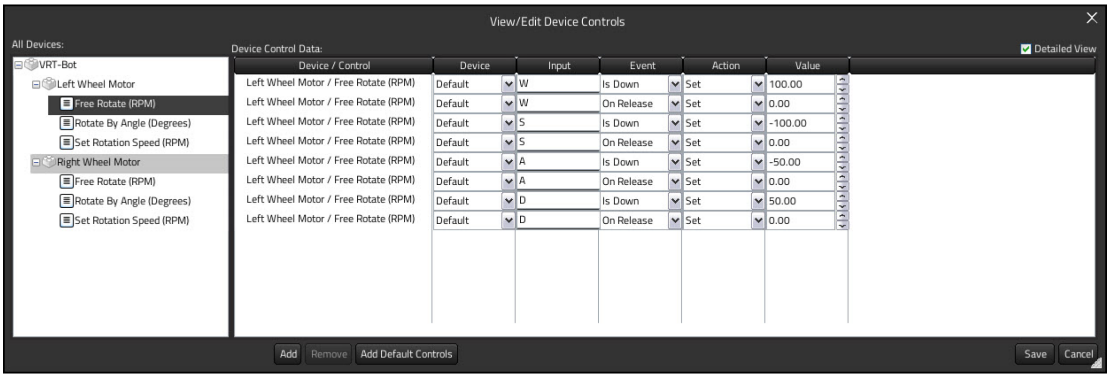


Figure 6.8: Device Controls dialog

6.4 Actor menu

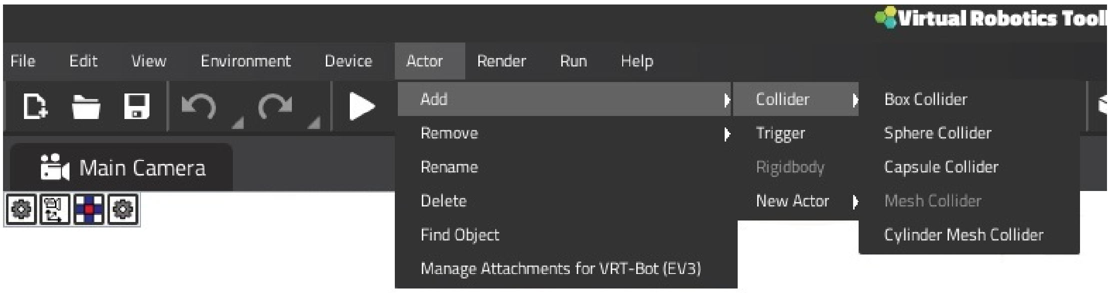
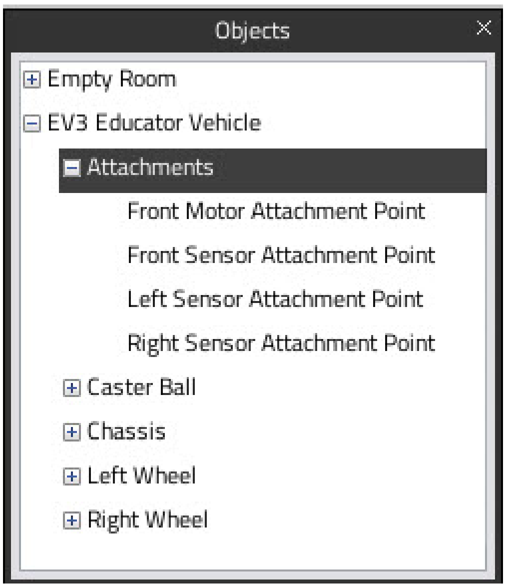


Figure 6.9: Actor menu

Actors represent the "things" or objects that the simulator must keep track of, and can contain one or more components. For example, a robot (actor) may have various motors or sensors (components) assigned to it. An actor's components determine the role that it will play in the simulation.

Actors are organized in a hierarchical fashion, that is, every actor can have a parent and any number of children. Actors that share a common parent are assumed to behave as a group, for that reason, position and rotation coordinates are relative to the parent.



For example, a robot will be typically represented by an actor with many children. In this case, each child corresponds to a specific sub-assembly of the robot, such as a left wheel, gear train, or a mechanical arm. Although these sub-assemblies will move with the robot, they can also have their own independent motions as well.

Through the Actor menu we can add, remove, or modify actors.

- Add: allows us to add a component or to create a new actor.
  - Collider: defines the shape of an object for the purposes of physical collisions. Properties relevant to collisions such as the physical material can be found in the **Collider** section of the **Object Properties** dialog.
  - Trigger: adding a trigger allows an actor to fire events, such as incrementing a score or setting a timer. A trigger is used in tandem with a collider. Events will fire whenever a collision occurs.
    - A trigger can be changed to behave as a region. This allows objects to pass through it, firing events whenever objects enter or leave that region. In this case, the collider forms the bounds for the trigger area.
  - Rigidbody: adding a rigidbody component to an actor identifies it as being an non-deformable object that will be affected by physics. This is where physical properties such as mass, inertia, and center of mass are defined. These attributes can be accessed via the **Rigidbody** section of the **Object Properties** dialog.
  - New Actor: creates an empty actor in the Objects window.
- Remove: allows you to remove both actors and components from the Objects window.
- Rename: allows you to rename an actor or component.
- Delete: allows you to delete an actor or component.
- Find Object: repositions the Main Camera to sit next to the selected object.

6.5 Render menu

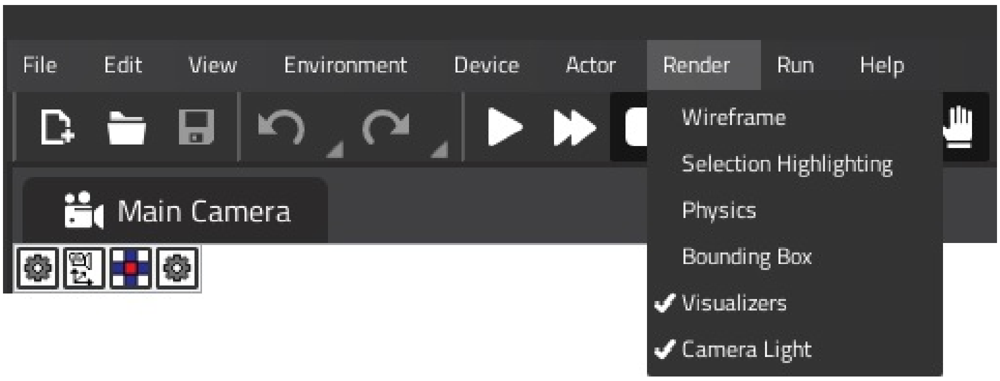


Figure 6.10: Render menu

The Render menu can be used to enable or disable various highlighting schemes for objects in the project file.

1. Wireframe: selecting this option will render the selected object as a skeletal three dimensional model in which only lines and vertices are represented.
2. Selection highlighting: will highlight the mesh of a selected object.
3. Physics: rendering physics, will show or hide the colliders and rigidbodies in the selected object. Review section 6.4 for a detailed explanation on how colliders and rigidbodies work.
4. Bounding box: is the area that can be clicked on to select an object.
5. Visualizers: are any effects produced by attached components such as the ping from an ultrasonic sensor.
6. Camera light: toggles between turning the light on the Main Camera on or off. In Model Space, the region on screen where the simulation occurs, we're actually viewing the environment from the perspective of a camera with an attached light. If this camera light is turned off, objects in the Model Space will only be illuminated by light sources in the environment.

6.6 Summary

As you can hopefully appreciate, there's a lot happening under the hood of this software. While the simulator is indeed a powerful tool, most of this added customization is not necessary in order to run a MINDSTORMS® simulation.
If you are interested in working with more complex types of models, or with robots outside of the LEGO® MINDSTORMS® platform where such considerations as altered physics and environmental settings are necessary, we'd love to hear from you.

Email us with your questions at support@cogmation.com

This concludes the written portion of the softwares' support materials, for additional video tutorials be sure to visit us on Youtube at:

`http://www.youtube.com/user/virtualrobotics`

Remember, only **YOU** can tell us how to have a good time!

Chris Schulz & the Cogmation team
December 17th, 2014