SECTION 5

ROBOT DESIGN SOFTWARE

ROBOT DESIGN METHODOLOGIES

Tools and Applications for Robot Design

Introduction

In this section, you will discover the tools and software that help you design and test robots. You will explore IDEs, modelling, and simulation tools, which are important for developing robotics projects. You will learn how to design robots in a virtual environment and see how they perform. You will be introduced to CAD software, where you will create models of parts for robotic systems. Finally, you will convert your CAD models into G-code and print them using a 3D printer.

At the end of this section, you will be able to:

- Explore features of selected modelling, programming and simulation tools useful for the design of robots.
- Design robots using virtual platforms and use simulation tools and programming IDEs to test the mechanics of the designed robots.
- Use a CAD tool to model parts of robotic systems.
- Use relevant intermediate tools to prepare modelled files into G-codes and print the designs using a 3D Printer.

Key Ideas:

- **Integrated Development Environments (IDEs):** Software that helps you write, test, and fix code all in one place.
- **Modelling Tools:** Used to create visual designs of robot structures.
- **Simulation Tools:** Let you test how robots will behave in a virtual setting before trying them out in real life.
- **3D Printing:** A method of printing which converts digital designs into three dimensional physical objects.
- **CAD (Computer-Aided Design):** Software used to design 3D models.
- **G-Code:** A type of code that gives instructions to 3D printers.

EXPLORING MODELLING, PROGRAMMING, AND SIMULATION TOOLS & APPS FOR ROBOT DESIGN

Integrated Development Environments (IDEs), modelling, and simulation tools are essential for developing robotics projects. Learning how to use these tools will help you design, program, and simulate robots more efficiently. This section will introduce you to the features of these tools and show you how they are used in robotics.

1. Integrated Development Environments (IDEs) for Robotics

IDEs are software packages which enable you to manage your robotics projects all in one place. Think of them as a one-stop shop for writing, testing, and fixing your robot's code so it works the way you want it to.

Here are some benefits of using IDEs:

a. **Code editing:** Write your code following language rules, using highlighting (colour-coding for easier reading) and auto-completion (suggestions as you type).

b. **Compiling and debugging:** Quickly turn your code into instructions the robot understands, and easily find and fix errors.

c. **Simulation and testing:** Some IDEs let you test your robot's behaviour in a virtual environment before trying it on real hardware.

Popular IDEs for robotics include:

a. **LEGO® Education Spike:** This tool helps you write programs for the Lego Spike Robot. It includes lessons, building instructions, and coding experiences that go from basic icon-based coding to more advanced text-based coding using Python.

b. **EV3 Classroom:** A user-friendly tool for programming LEGO® EV3 kits with an intuitive icon-based environment that connects to the EV3 robot.

c. **Arduino IDE:** A widely used tool in robotics for programming Arduino microcontrollers. It has an easy-to-use interface for writing, compiling, and uploading code.

d. **Visual Studio Code (VS Code):** A more advanced IDE that supports multiple microcontroller platforms, offering strong code editing and debugging features.

e. **MATLAB/Simulink:** MATLAB is powerful for data analysis and Simulink is often used for modelling and simulating robotic systems.

2. Modelling and Simulation Tools for Robotics

Modelling and simulation tools are like virtual playgrounds for building robots. They let you create a digital version of your robot in an environment, then test how it moves and behaves before trying it out in the real world. These tools help you visualise and test robotic systems before building them.

Benefits of using modelling and simulation tools:

a. **Safer testing:** Experiment with different programming approaches without risking damage to your robot.

b. **Faster development:** Find and fix errors early in the process, saving time and resources.

c. **Optimised performance:** Test different settings to find the most efficient way for your robot to operate.

d. **Visualise robot behaviour:** See how your robot interacts with its environment before building it.

Popular tools for modelling and simulation include:

a. **BrickLink Studio:** A powerful 3D modelling software designed for creating virtual LEGO® models. It has an extensive library of LEGO® parts that is regularly updated.

b. **ROS (Robot Operating System):** An open-source framework that provides tools for developing robotic applications, including simulation and communication between robot components.

c. **Gazebo:** A robust 3D robot simulation environment that works well with ROS, allowing users to test robotic algorithms in virtual environments.

d. **Webots:** A professional simulation software used to model, simulate, and optimise robotic systems.

e. **Tinkercad:** A simple software for creating 3D models of robots that can be exported for 3D printing.

3. Summary of differences between IDEs and Modelling / Simulation tools

You will find that some IDEs share similar features with modelling and simulation tools. The differences between the two groups are important for robotics engineers who will choose a tool for a purpose. These are:

a. Focus: IDEs focus on software development; while modelling and simulation tools focus on testing and validating robot designs and behaviours.

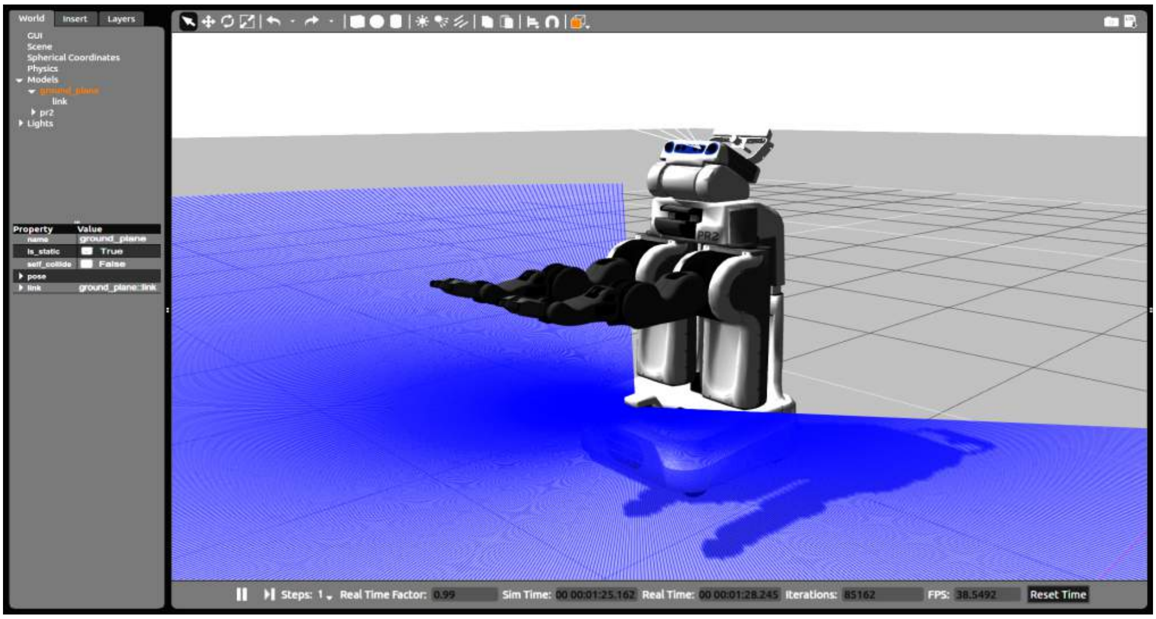
b. Environment: IDEs provide a coding environment, whereas modelling and simulation tools provide a virtual environment for testing.

c. Output: The primary output of an IDE is executable code, while the output of a simulation tool is data on the robot’s performance and behavior in simulated scenarios.

4. Features of Selected IDEs (Programming) Modelling and Simulation Tools#

In this section you will find out how these tools work: ROS and Gazebo, Tinkercad, the Virtual Robotics Toolkit, and Arduino IDE. The links will help you to understand how to operate each tool.

a. **ROS and Gazebo:** ROS provides a modular system that makes it easy to connect different parts of your robot, while Gazebo gives a realistic virtual environment to test your robot's movements and interactions.



Property	Value
name	ground_plane
is_static	☑ True
self_collide	□ False
pose	
link	ground_plane::link

Steps: 1 Real Time Factor: 0.99 Sim Time: 00 00:01:25.162 Real Time: 00 00:01:28.245 Iterations: 85162 FPS: 38.5492 Reset Time

Fig. 5.1: ROS Simulation

Scan the QR code below for an overview of ROS and Gazebo, its installation instructions as well as some extended features:

Link	QR CODE
https://www.youtube.com/watch?v=laWn7_cj434	

b. **Tinkercad**: Tinkercad is a free software package. It's an easy-to-use web-based app. You can start using it right away without any downloads or strings attached. Tinkercad can be used for creating 3D models, simulating electronic circuits, and 3D printing. It is a flexible tool which can be used to adapt your designs and test them in a virtual environment before actually building your project.

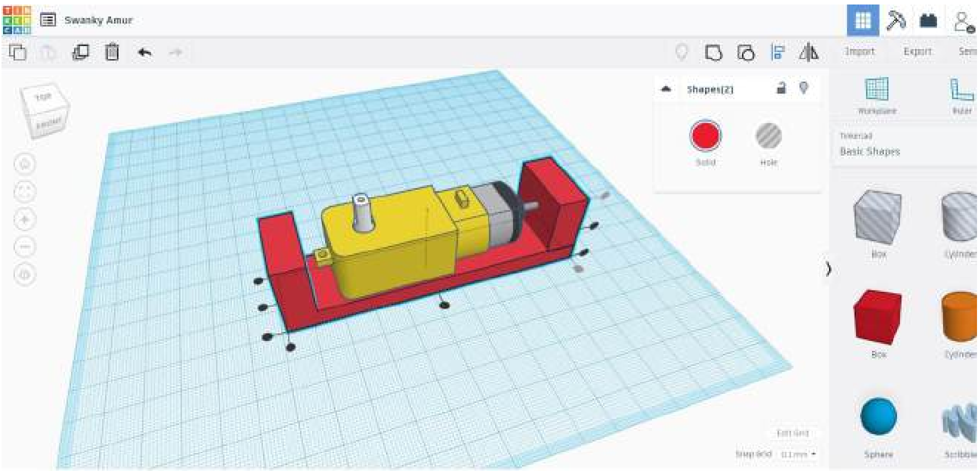


Fig. 5.2: Tinkercad

Scan the QR code below for an overview of Tinkercad and how to create your design.

Link	QR CODE
https://www.youtube.com/watch?v=9hkbGngpGVk	

c. **The Virtual Robotics Toolkit**: The Virtual Robotics Toolkit is software that lets you learn and practice building robots in a simulated environment. You can use it to work on robotics projects without needing to build a robot using component parts.

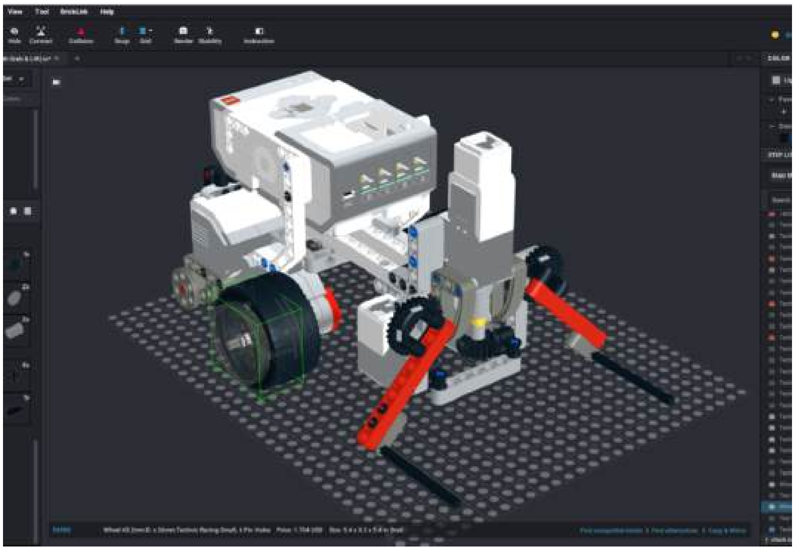
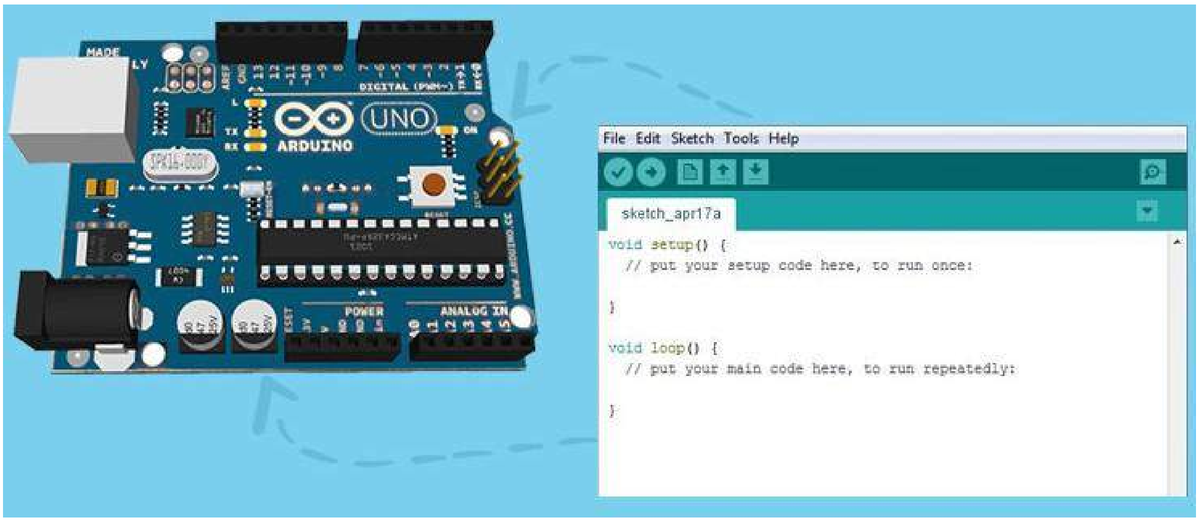


Fig. 5.3: Virtual Robotics Toolkit

d. **The Arduino IDE:** The Arduino Integrated Development Environment (IDE) is a powerful tool for robotics projects. With the Arduino IDE, you can write, compile, and upload code to Arduino boards, which are used to control various components like motors, servos, and sensors. This allows you to build and program robots that can perform a wide range of tasks, from simple movements to complex operations.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fig. 5.4: Arduino IDE

Scan the QR code below for an overview of installation and usage of the Arduino IDE.

Link	QR CODE
https://www.youtube.com/watch?v=Wkt28joEJUE	

Activity 5.1: Installing the Arduino IDE

Scan the QR code below for the step-by-step guide through the installation process of the Arduino IDE on your preferred operating system (Windows, macOS, Linux) and follow the steps shown in the video to download and install the Arduino IDE on your computer.

Link	QR CODE
https://www.youtube.com/watch?v=3awCkLS7gHI	

Answer the following questions after you have ensured that the IDE is running without errors.

1. What were the steps involved in installing the Arduino IDE?
2. Did you encounter any issues during the installation process? How did you resolve them?
3. Why is it important to install the correct drivers for your operating system?

Activity 5.2: Exploring the features of the Arduino IDE

Scan the QR code below for the video that provides a general overview of the Arduino IDE's features, including the toolbar, code editor, serial monitor, and libraries.

Link	QR Code
https://www.youtube.com/playlist?list=PLwWF-ICTWmB7-b9bsE3UcQzz-7ipI5tbR	

Open the Arduino IDE and explore the different sections highlighted in the video. Some key areas to explore:

1. Code Editor: Write a simple "Hello World" program (e.g., blink an LED).
2. Toolbar: Identify the buttons for verifying and uploading code, opening serial monitors, and managing libraries.
3. Serial Monitor: Test the Serial Monitor by sending a basic message from the Arduino to your computer.
4. Libraries: Explore how to include and manage libraries for different components like sensors or motors.

Activity 5.3: Exploring the features of the Arduino IDE

1. What are the key features of the Arduino IDE that you discovered during your exploration?
2. How do you compile and upload a program to an Arduino board using the IDE?
3. Describe how the Serial Monitor works and provide an example of when it would be useful.
4. How do you add a new library in the Arduino IDE? Why might you need to use libraries in your projects?
5. What is the purpose of the verification (checkmark) button in the toolbar?

Activity 5.4: Exploring Tinkercad Features

Scan the QR code below for videos that provides a general overview of the features of Tinkercad including 3D design, Circuit design and simulation and code blocks.

Link	QR CODE
https://www.youtube.com/watch?v=9RF_BZ1Cg4k	
https://www.youtube.com/watch?v=PxGNnoLXgcQ	

Follow the instructions in the video which gives an overview of Tinkercad and create your own account. Explore the features of Tinkercad:

a. **Exploring 3D Design:**

   i. Start a new 3D design project and use basic shapes (e.g., cubes, cylinders, spheres) to create a simple model, such as a keychain, a nameplate, or a mini robot.

   ii. Practice modifying the shapes by resizing, rotating, grouping, and aligning them.

   iii. Use the “Hole” feature to subtract shapes and create cutouts in your design.

b. **Exploring Circuit Simulation:**

i. Switch to the Circuits workspace.

ii. Build a basic circuit, such as an LED connected to a resistor and a battery or a push-button circuit.

iii. Run the simulation to see how your circuit works. Experiment with changing the values of components like resistors to see the effects.

iv. Explore and add more components, like motors or sensors, to make your circuit more complex.

Share your observations and experiences from Activities 5.1-5.3 with your class.

VIRTUAL ROBOT DESIGN AND SIMULATION: EXPLORING MECHANICS AND TESTING

Testing the mechanical actions of robots using virtual platforms and simulation tools involves creating a digital model of the robot and its environment. In this virtual space, you can simulate how the robot's parts move and interact without needing to build a physical prototype. This process helps you identify and fix any issues in the design, optimise the robot's performance, to ensure that it behaves as expected in different scenarios. By using simulation tools, you can save time and resources while gaining a better understanding of how your robot will function in the real world.

1. Virtual Platforms for Robot Design:

Virtual platforms are computer programs that let you design and interact with simulated robots. These platforms offer features like 3D modelling, sensor and task simulations, control and risk testing, considering the effect of real-world physics like material properties or gravity and visualisation.

Virtual platforms for robot design include:

a. **Bricklink Studio**: Bricklink Studio is software based on LEGO® that provides a flexible environment for creating, visualising, and simulating complex robotic systems. BrickLink Studio, often just called Studio, is an all-in-one desktop application for Windows and Mac that allows you to build with virtual LEGO® parts. BrickLink Studio can be used to design and visualize robotic models. However, it doesn't offer advanced simulation features like physics-based movement or sensor integration that are typically required for testing robotic actions.

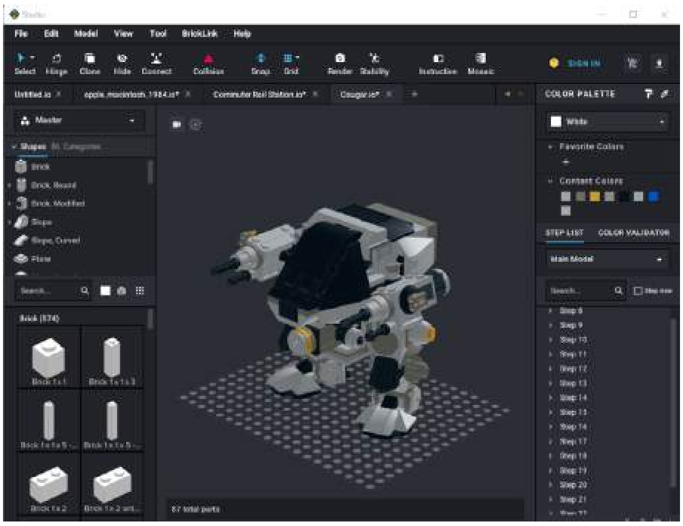


Fig. 5.5: Bricklink Studio Interface (Simulation). (Fileinfo, 2021)

Scan the QR Code below for an overview of Bricklink Studio

Link	QR Code
https://www.youtube.com/watch?v=TCDMvVfd86o&t=15s	
https://www.youtube.com/watch?v=BUxP1A4NLGE	

b. **Tinkercad**: You should know already that Tinkercad is a free software package. It's an easy-to-use web-based app for 3D design, electronics, and coding. You can download it and start using it on your own computer, if you have a connection to the Internet. Tinkercad can be used for designing and simulating the mechanical actions of robots. You can create 3D models of robotic parts, program robots using block-based coding, and export designs for 3D printing using Autodesk Fusion.

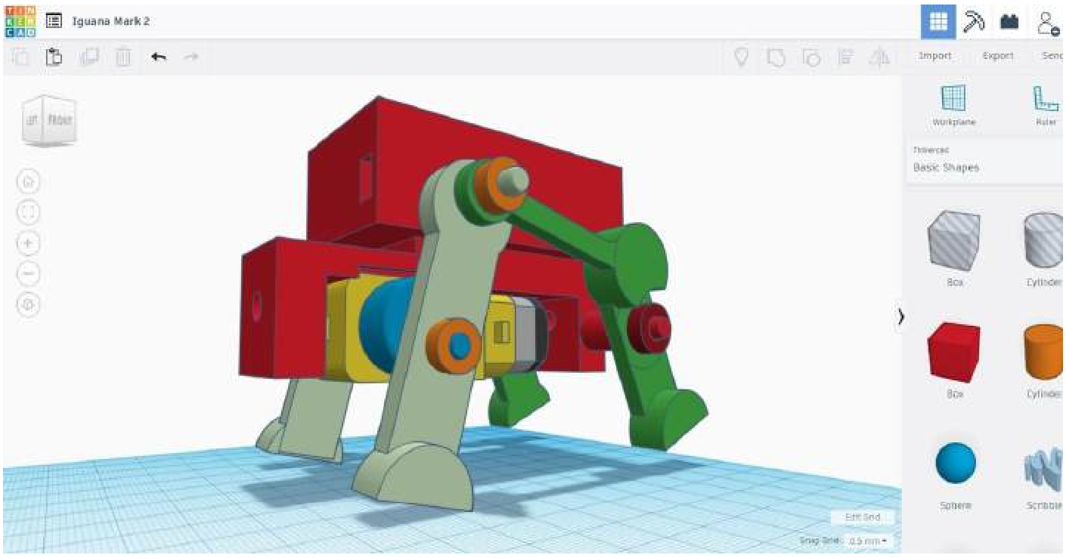


Fig. 5.6: Tinkercad Interface (Tinkercad, 2020)

Tinkercad can be used to run simulations to test your projects before you build them.

Try this using Tinkercad: Build a simple circuit to light up an LED when a switch is pressed.

Step 1: Add Components: Drag and drop the components you need from the **library** onto the **workspace**. This could include resistors, LEDs, batteries, switches.

Step 2: Connect Components: Use the **wiring tool** to connect your components as needed. Make sure all connections are correct to avoid errors in the simulation.

Step 3: Switch to **Simulation Mode**: Click on the “Code” button at the top right of the screen, then switch to the “Blocks” or “Text” tab to write your code if needed.

Step 4: Start Simulation: Click the “**Start Simulation**” button at the top right. This will run your circuit and show you how it behaves.

Step 5: Monitor and Adjust: **Observe the simulation results**. If something is not working as expected, stop the simulation, make adjustments, and run it again.

c. **Virtual Robotics Toolkit**: The Virtual Robotics Toolkit is a simulator tool which uses the LEGO® Mindstorms EV3 programming environment to control the mechanical actions of virtual robots. It lets you control a virtual robot without needing space for physical testing or worrying about running out of LEGO® bricks.

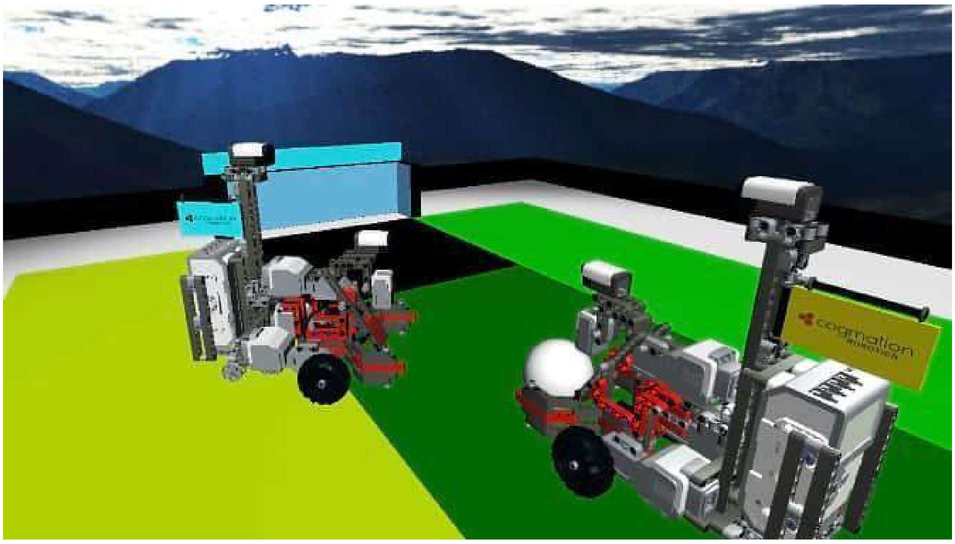
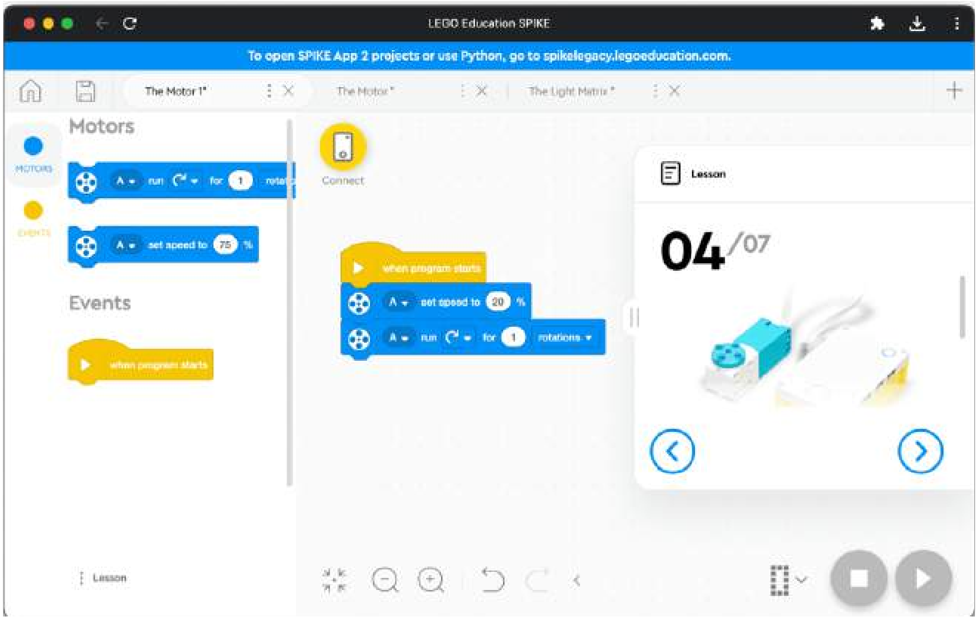


Fig. 5.7: Virtual Robotics Toolkit

2. Exploring Robot Mechanics using LEGO® Education SPIKE App (IDE):

Integrated Development Environments (IDEs) are used in projects to program robot mechanical actions . The LEGO® Education SPIKE App gives you an easy and enjoyable way to program the mechanical actions you want your robot to make. Once you have designed and built your virtual robot you can use this platform to program it using icon-based blocks, word-based blocks, or text-based coding. You do not have to know how to write code and the app displays error messages to help identify and correct mistakes. The app is free to download and use but remember it is only designed to work alongside other coding apps from LEGO® Education.



To open SPIKE App 2 projects or use Python, go to spikelegacy.legoeducation.com.

Fig. 5.8: LEGO® Education SPIKE App (IDE)

It also provides users with tutorials to start building robot structures using the LEGO® Education SPIKE™ Essential or LEGO® Education SPIKE™ Prime Kits. Figure 14.5 shows a snapshot of the interface for doing this.

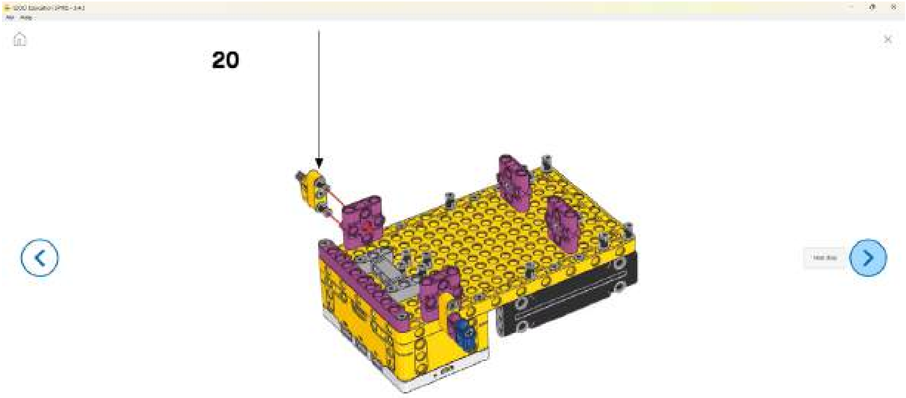


Fig. 5.9: LEGO® Education SPIKE App Build tutorials for constructing robots

Also, to help users with the basics of programming these kits, it provides some basic tutorial activities. These activities cover the major controllers, sensors and actuators.

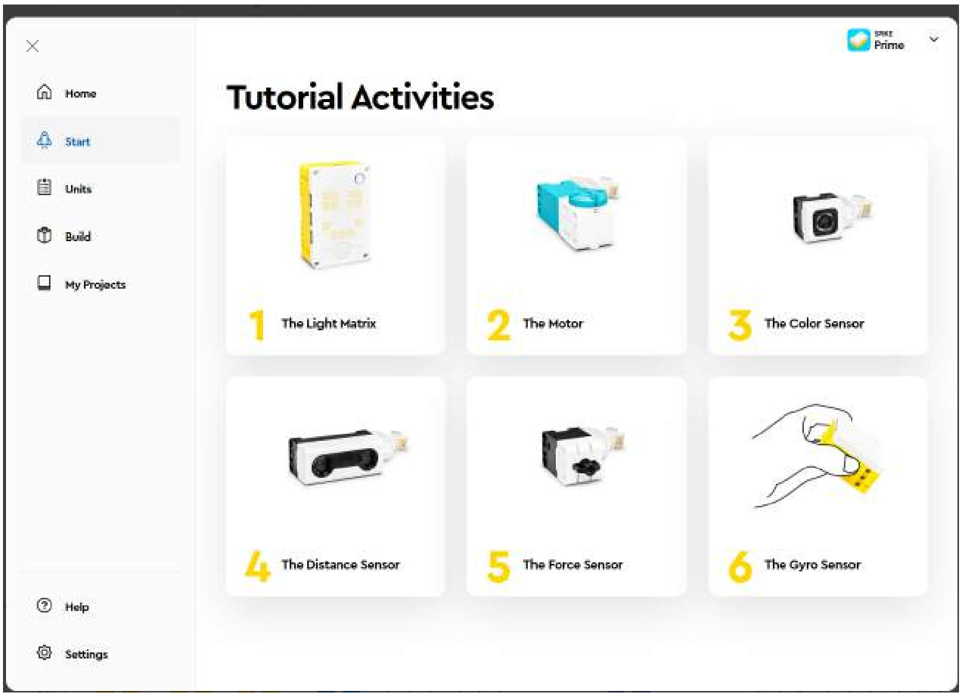


Fig. 5.10: Tutorial Activities in LEGO® Education SPIKE App

Scan the QR code below for a video on how to download and install LEGO® Spike App to your computer.

Link	QR Code
https://www.youtube.com/watch?v=Cqgl9pF0Us0	

3. Simulating Robot Performance is valuable:

Simulations provide a valuable tool for testing and evaluating robot performance without the need for physical prototyping. The main advantage of a simulation is that it is a cost-effective way of achieving the best possible performance. In virtual environments it is possible to:

a. **Test Algorithms**: Implement and test control algorithms for navigation, path planning, and obstacle avoidance in a risk-free setting.

b. **Validate Design Choices**: Evaluate the robot's stability, agility, and efficiency based on its virtual performance, leading to improved design decisions.

c. **Error and Risk Analysis**: Analyse potential errors or risks that the robot may encounter during real-world operation, helping with problem-solving.

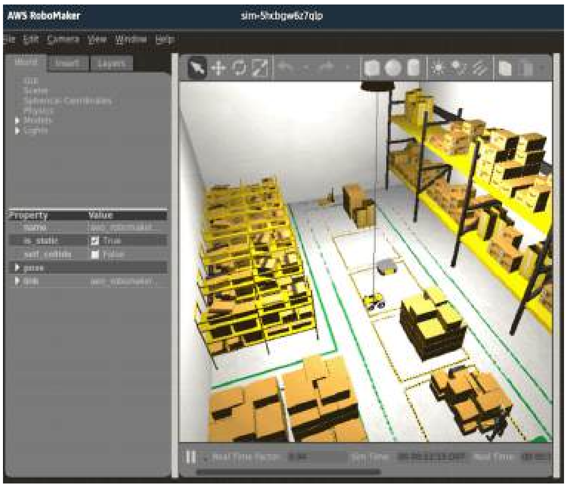


Fig. 5.11: Virtual simulation of a mobile robot navigating through a warehouse environment (Vamshi Konduri, 2020).

Activity 5.5

Your teacher will divide you into groups for this activity. Your group should discuss the virtual platforms that can be used to design, build, program and test robot mechanical actions. Based on the provided information (above) and your experience from investigating the online resources your group should **identify which platform offers the best experience**.

Your group should create a comparison chart for each platform which assesses three things; the user friendliness of the interface (e.g., was it easy or hard to learn how to use it?), what is supported by the platform(e.g., motor and sensor types, physics properties, 3D modelling, visualisation, task simulation), and suitability for beginners (e.g., examples provided, menus and pathways clear, instructions section, help menu, visualisations) (**NOTE**: This chart will be presented to the class).

The reasons for your choice of which platform offers the best experience should be in the form of a written summary on your chart.

Activity 5.6

In your groups, use your knowledge of sensors, controllers and actuators, video resources and previous tasks to design a robot capable of navigating the spaces between shelves in a simulated warehouse environment using Tinkercad.

Activity 5.7

For the robot your group designed to work in a warehouse environment run simulations using Tinkercad to test robot performance and identify potential issues. Refine the design based on the simulation results.

CAD MODELLING AND 3D PRINTING FOR ROBOTIC SYSTEM PARTS

In the following information, you will explore the Computer-Aided Design (CAD) and 3D printing of robotic system parts. CAD is used to design precise components that can be made into physical objects using a 3D printer. 3D printing has changed the way mechanical parts are manufactured, allowing design freedom, unique build, reduced waste, lightweight components and faster production. You will learn the steps involved in CAD and 3D printing for the creation of custom parts for robotic systems.

Introduction to CAD Modelling:

Computer-Aided Design (CAD) uses software to create detailed 2D or 3D models of objects. CAD modelling is essential for designing robotic parts with precision and complexity. CAD software allows us to see, change, and test designs before printing them.

CAD Tools for Robotics:

There are several CAD software options available for designing robotic parts. Some of the most common include:

a. **Auto desk Fusion 360**: Fusion 360 is a powerful CAD tool that combines 3D modelling, simulation, and teamwork features. It's widely used in robotics and engineering due to its easy-to-use interface.

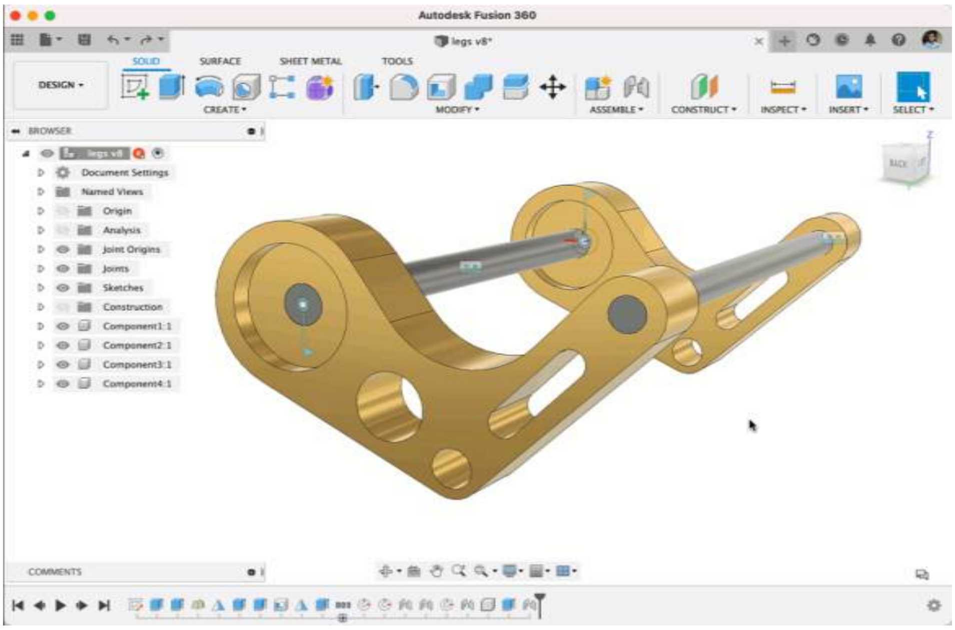


Fig. 5.12: Autodesk Fusion 360 User Interface

b. **Solid Works**: SolidWorks is a popular CAD software known for its advanced modelling features and specialised tools for different industries. It's great for designing complex robotic parts.

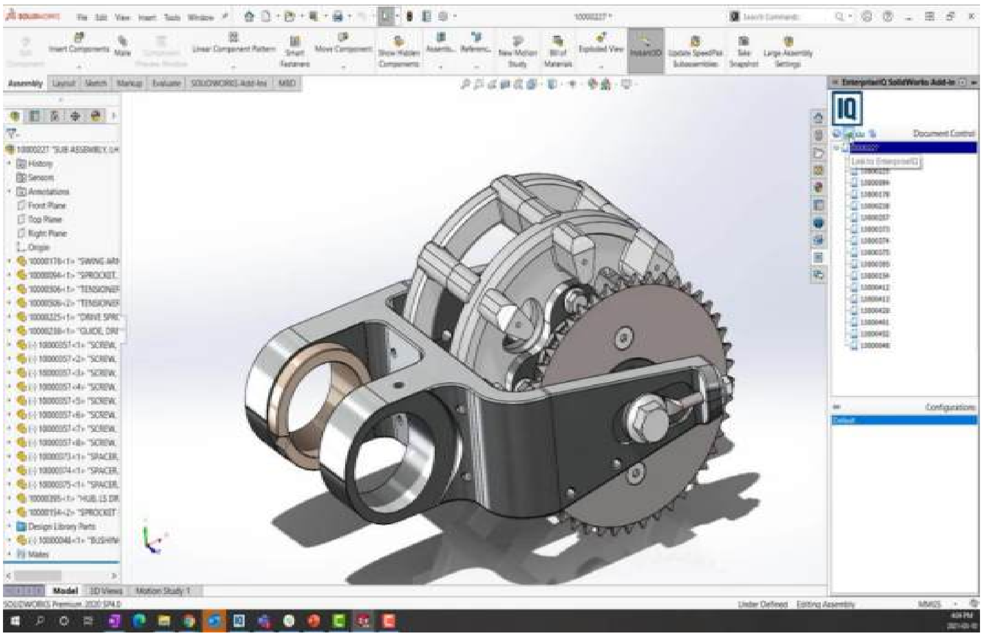


Fig. 5.13: Solid Works User Interface (Rob Hauser, 2021)

c. **Tinkercad**: Tinkercad is a CAD tool that is suitable for beginners. You can use it as a starting point for learning the basics of CAD modelling.

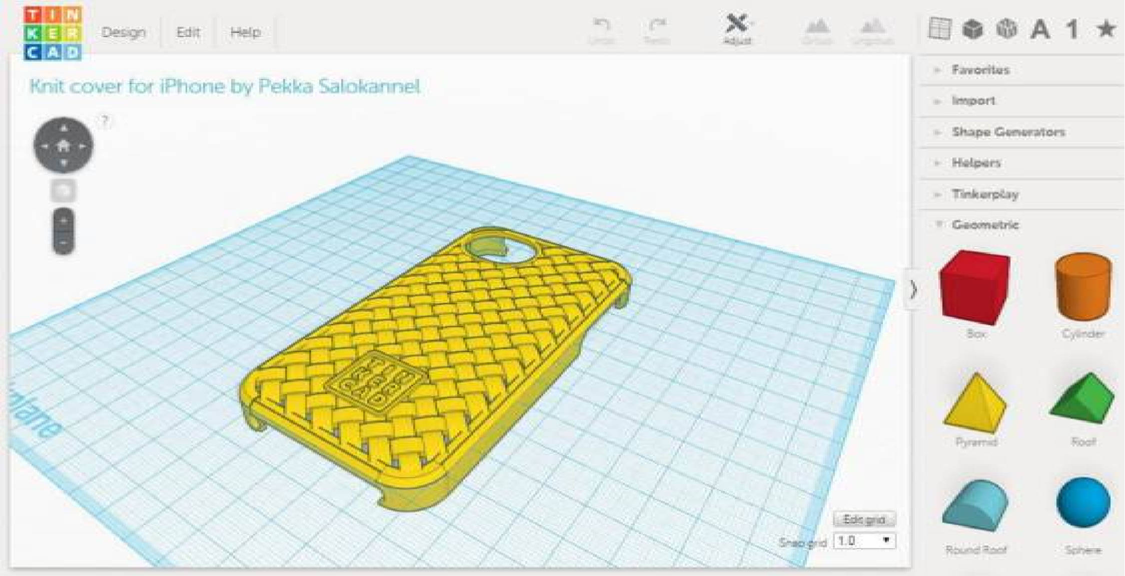


Fig. 5.14: Tinkercad Interface

CAD Modelling of Robotic Components using Tinkercad

The following information is to help you design a chassis and four wheels for a moving robot using Tinkercad as your choice of CAD software. You can ask your teacher for help if you need it.

a. **Robot Chassis:**
The chassis is the frame to which all the robot's parts are fixed. Use your CAD software to design a chassis, considering the structural strength, size, shape, and spots for attaching sensors, wheels and motors.

b. **Wheel Assembly:**
Use your CAD software to design the wheel assembly. Include how many will be driven by the motor(s). Consider factors like traction (grip), size, and how the wheels will be attached to the chassis and/or motors.

Scan the QR Code below for a Tinkercad video which will help you design the two components for your own 4 Wheel Chassis.

Link	QR Code
https://www.youtube.com/watch?v=-UhGb81V-Fo	

Exporting for 3D Printing:

After finishing your two designs using your CAD software, you will need to prepare them for 3D printing. To do this you need to save and export your designs in a file format that works with the 3D printer, like STL (stereolithography) or OBJ (Wavefront OBJ). Designs exported from Tinkercad in formats like STL or OBJ need to be imported into slicing software which will generate a G-code before they can be 3D printed. G-code is the language that 3D printers understand, containing instructions on how to move the printer's components to create the object layer by layer (or in slices).

3D Printing the Robotic Components:

First task is to get the 3D printer ready by selecting the right printing settings, like layer height and print speed. This depends on the size and complexity of your design. Load your CAD files into the 3D printer software (which will convert your CAD files to G-code) and start the printing process. Watch as your designs are built layer by layer.

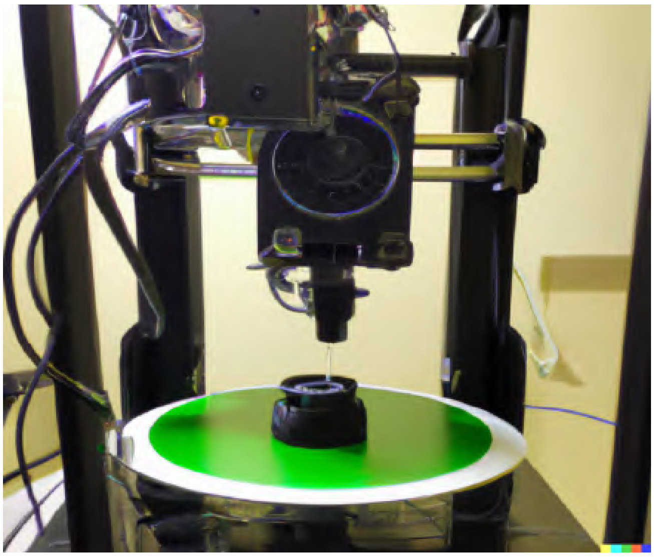


Fig. 5.15: 3D Printing Process

Scan the QR Code below for an overview of Creating a 3D Printed Object: from Digital Concept to Physical Object (Tinkercad to Ender 3 Pro)

Link	QR Code
https://www.youtube.com/watch?v=7xKe20Mb2eY	

Activity 5.8

Working in a small group, design a robotic component using Tinkercad. This could be a gripper or claw, part of a robot arm or chassis. Sketch your ideas first, then translate them into a 3D model using the Tinkercad software.

Present your robotic component design to another group. Explain the purpose of the component, your design process, and any other things you took into consideration. Listen to their feedback and be prepared to make any changes to your design.

Activity 5.9

Working with your group, refine your robotic component design based on feedback. Optimise your design by making sure you choose an appropriate material for your printed component and check there will be no problems with structural strength during the printing process like sections that are too thin or any necessary support structures are in place. Prepare your design for 3D printing by saving it to an appropriate file such as an STL file.

Activity 5.10

Working alone, write a reflection of 300-500 words that captures your learning journey through the above activity (What challenges did you face while using Tinkercad?) How did your design change after feedback, and what did you learn about optimising designs for 3D printing?).

3D PRINTING WITH G-CODES: FROM CAD MODELLING TO PHYSICAL PROTOTYPES

The information below describes in detail the final steps of producing a physical model from design idea to 3D printing. It will describe the intermediate tools which are used to prepare your CAD (Computer Aided Design) files from which G-codes can be generated to print your designs using a 3D printer. Understanding the role intermediate tools play in the preparation for conversion to G-codes and printing is important for producing an error free physical model.

1. Preparing CAD Models for 3D Printing:

Before printing, CAD models need to be checked and prepared. This includes:

a. **Model Inspection:** Make sure the CAD model is error-free, with no overlapping or intersecting parts.

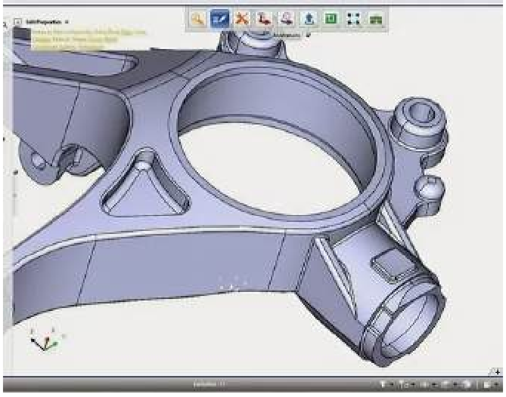


Fig. 5.16: 3D printing-model inspection

b. **Scale and Orientation:** Adjust the size and position of the model to fit the 3D printer's build area and improve printing speed and quality.

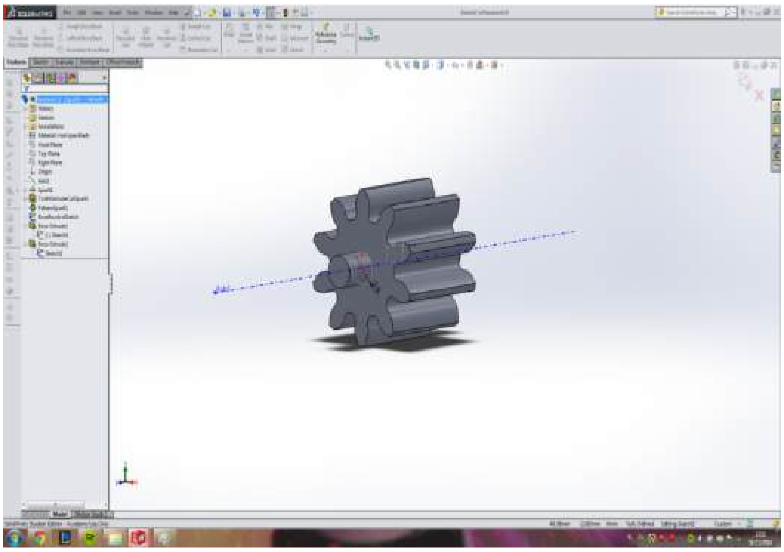


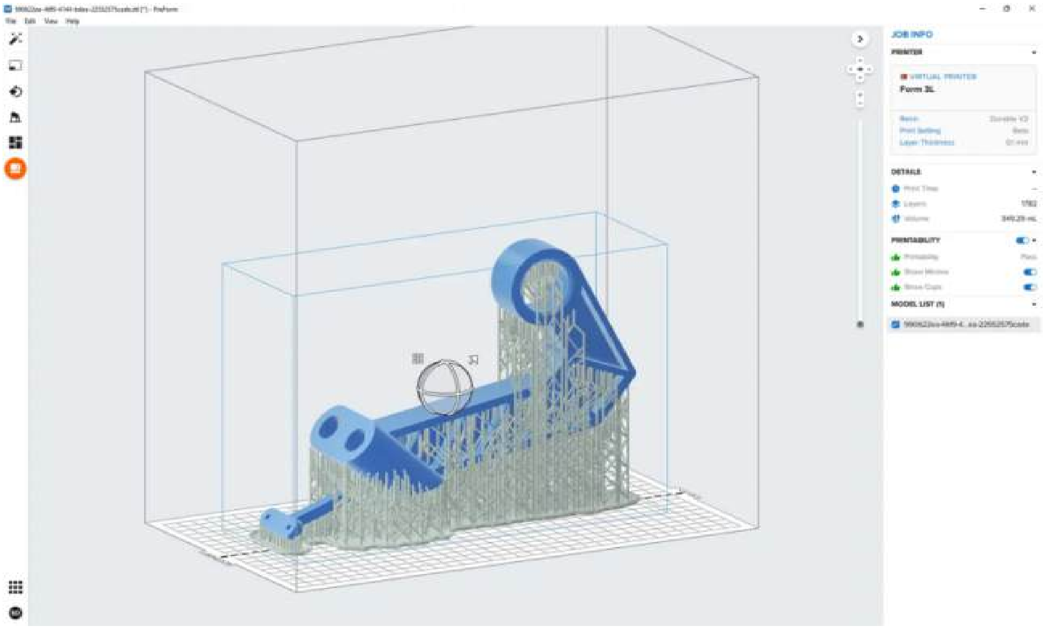
Fig. 5.17: 3D printing- scale and orientation

c. **Supports and Rafts:**

These elements help ensure successful 3D prints by providing stability and support during the printing process.

Support provides temporary structures to support overhanging parts of a print that would otherwise collapse. They can be generated automatically by slicing software and are usually made of the same material as the print. They are removed after printing, often requiring some post-processing to smooth the surface where supports were attached.

Rafts create a flat, stable base for the print to improve bed adhesion and prevent warping (twisting out of shape). Rafts consist of several layers of filament laid down before the actual print begins. They are useful for small items or materials prone to warping. They are detached from the final print, sometimes leaving a rough surface that may need smoothing.



Layers 1782
Volume 349.29 mL

Fig. 5.18: 3D printing-Support and Raft

2. Intermediate Tools: Slicers for G-Code Generation:

A slicer is software tool in 3D printing is a software program that takes a 3D model and slices it into thin layers. It then generates the necessary instructions, known as G-code, for a 3D printer to create the physical object.

The following are slicer tools:

a. **Ultimaker Cura**: This is a slicer software program with a simple interface and advanced features. It supports many 3D printers and gives detailed control over printing settings.

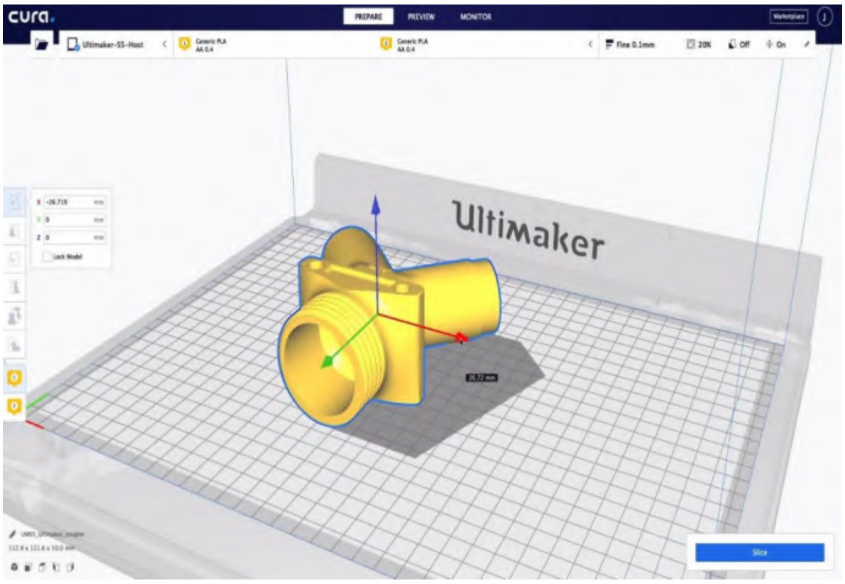


Fig. 5.19: Slicing process using Ultima Cura

Scan the QR code below for videos on an overview of Ultimaker Cura

Link	QR Code
https://www.youtube.com/watch?v=cUVEoGfN45g	
https://www.youtube.com/watch?v=eUNTlb5pEWA	

b. **PrusaSlicer**: PrusaSlicer software is designed for Prusa 3D printers only and integrates smoothly with their hardware. It also offers advanced customisation for printing.

c. **Simplify3D**: Simplify3D is a flexible slicer tool that provides a lot of control over print settings. It works with many different 3D printers and is popular with experienced users.

3. Exporting G-Codes and Printing:

Once the model is sliced and the G-codes have been generated, start the 3D printing process by following these steps:

a. **Export G-Codes**: Save the G-code file to an SD card or transfer it to the 3D printer using a USB connection.

b. **Preparing the 3D Printer**: Ensure the printer is properly set up, with the bed levelled and the correct filament (construction material) loaded.

c. **Start Printing**: Insert the SD card with the G-code or send the print command from the 3D printer's interface. The printer will begin creating the object based on the G-code instructions.

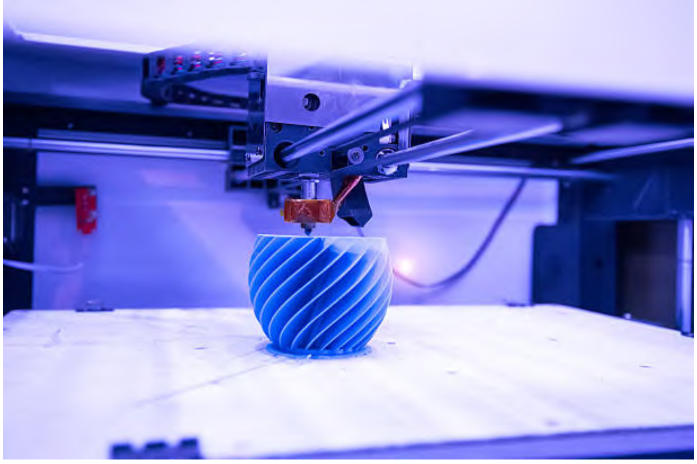


Fig. 5.20: 3D Printing Process in Action

Scan the QR code below for an overview of model File to 3d Print - A Beginners Guide to Using Ultimaker Cura (3d Printer Slicer Software)

Link	QR Code
https://www.youtube.com/watch?v=e-lQPGJ3Psc&t=350s	

NOTE: You may work on your own for the following activities or your teacher will organise you into your groups. Work with your colleagues to ensure everyone is able to participate in the activities.

Activity 5.11

Inspect your chosen CAD model using Tinkercad's analysis tools. Look for common issues such as thin walls or overhangs that may need support. Make any necessary adjustments to your design to ensure it's ready for 3D printing.

Submit a revised model with a brief explanation of the adjustments made and why they were necessary.

Activity 5.12

Import your chosen CAD model into Ultimaker Cura 3D Slicer. Configure the slicing settings, such as layer height and infill percentage, based on the complexity of your model. Generate the G-code that will control the 3D printer. Save your files ready to export to the printer.

Write a short explanation of how your slicing settings affect the print quality and speed.

Activity 5.13

Load your G-code file for your CAD model onto the 3D printer software and begin the printing process. Monitor the printing of our design, If you notice any problems with the printing inform your teacher or supervisor.

a. Write notes while observing the printing process, including recording any problems you had and how you resolved them. Take pictures or videos of the print process.

b. Complete a written evaluation of your printed model and remove any filling or unwanted materials. Consider things like quality of print, whether the model as fit for purpose and what you would have done differently to improve the output.

NOTE: For this activity your school must have a 3D printer and a supply of filament (material used to printing your object) If you don’t have one in your school, your teacher should be able organise to visit a school or institution (e.g., Senior High Schools, Universities, Technical Universities or even a private company) that has a 3D printer. Save your files on portable drive like a pen drive or memory stick to take with you to print your object.

Review Questions 5.1: Programming, Modelling and Simulation Tools

1. Which of the following components is NOT part of the Arduino IDE?
   a. Code Editor
   b. Toolbar
   c. Electronic Circuit Simulator
   d. Serial Monitor

2. What does the checkmark button in the Arduino IDE toolbar do?
   a. Uploads the code to the Arduino board
   b. Verifies/compiles the code for errors
   c. Opens the Serial Monitor
   d. Saves the project

3. Where can you manage external libraries in the Arduino IDE?
   a. Serial Monitor
   b. File Menu
   c. Sketch Menu
   d. Preferences

4. Which feature in Tinkercad allows you to simulate an electronic circuit?
   a. 3D Design Workspace
   b. Code Blocks Workspace
   c. Circuit Workspace
   d. Shape Generator

5. In the 3D Design workspace of Tinkercad, what does the “Hole” tool do?
   a. Adds a new shape to the design
   b. Cuts out parts of a shape
   c. Changes the colour of a shape
   d. Combines two shapes into one

6. What are the main purposes of the robot modelling and simulation tool like Tinkercad?

7. Explain why modelling and simulation are helpful when undertaking robotic projects.

Review Questions 5.2: Virtual Platforms

1. Which of the following virtual platforms allows you to build and simulate robotic designs in a 3D environment?
   a. Bricklink Studio
   b. Tinkercad
   c. Virtual Robotics Kit
   d. All the above
2. Which feature would be most important for a virtual platform if designing a robot with multiple sensors controlling mechanical actions?
   a. User interface complexity
   b. Component library and sensor support
   c. Speed of the simulation
   d. Colour options of the robot parts
3. True or False: Tinkercad is a virtual platform that includes both electronic circuit and robotics design functionalities.
4. True or False: Simulation tools can help identify potential issues in a robot’s design before physical construction.
5. True or False: The LEGO® Education SPIKE App is a virtual platform that allows for 3D modelling and simulation of robots.
6. Identify three robotic actions that can be simulated in the software packages you have explored in section 5.

Review Questions 5.3: CAD tools

1. For a CAD software package you have used, state three design features that it includes.
2. Describe the primary purpose of the distinct features mentioned above.
3. Outline the procedure for preparing a CAD design of a robotic component for 3D printing.

Review Questions 5.4: 3D Printing

1. List two 3D slicing software packages
2. Why is a 3D slicing software needed to work on CAD models before printing?
3. True or False: PrusaSlicer is designed specifically for use with Prusa 3D printers.
4. True or False: The CAD model should be inspected for errors and adjusted for scale and orientation before generating G-codes.

5. What is the primary purpose of a slicer in the 3D printing process?
   a. To create CAD models
   b. To convert CAD models into G-codes
   c. To adjust the build volume of the 3D printer
   d. To print the physical prototype

6. Which of the following slicer software is known for its user-friendly interface and compatibility with various 3D printer models?
   a. PrusaSlicer
   b. Simplify3D
   c. Ultimaker Cura
   d. AutoCAD

Answers to Review Questions 5.1: Programming, Modelling and Simulation Tools

1. C. Electronic Circuit Simulator
2. B. Verifies/compiles the code for errors
3. C. Sketch Menu
4. C. Circuit Workspace
5. B. Cuts out parts of a shape
6. The main purposes of tools like Tinkercad are:
   a. 3D Modeling: Create custom parts and components for your robots.

   b. Circuit Simulation: Design and test electronic circuits, including those with microcontrollers.

   c. Prototyping: Build virtual prototypes of your robots to test their functionality before physical assembly.

   d. 3D Printing: Convert your designs into printable files to create physical parts using a 3D printer.

   e. Learning and Teaching: Use Tinkercad as an educational tool to learn more about robotics, electronics, and 3D design.

   f. Collaboration: Share your designs and collaborate with others on robotics projects.
7. Using modelling and simulation is helpful when undertaking robotic projects for several reasons including:
   a. Cost-Effective: Test designs virtually before building physical prototypes, saving money on materials.

   b. Time-Saving: Quickly identify and fix issues in the design phase, reducing the time spent on physical adjustments.

   c. Safety: Safely test potentially dangerous scenarios without risking harm to people or equipment.

   d. Optimization: Experiment with different designs and configurations to find the most efficient solution.

   e. Visualization: See how your robot will look and behave in a virtual environment before actual construction.

   f. Learning: Gain a deeper understanding of how different components interact and function together.

Answers to Review Questions 5.2: Virtual Platforms

1. D) All of the above
2. B) Component library and sensor support
3. True
4. True
5. False (The LEGO® Education SPIKE App is primarily an IDE for programming, not for 3D modelling and simulation)
6. Here are three robotic actions that can be simulated in the software packages described in section 5 (Bricklink Studio, Tinkercad, Virtual Robotics Toolkit, and LEGO® Education SPIKE app):
   a. Navigation and Pathfinding: Simulating a robot's ability to move through an environment, follow a specific path, or navigate around obstacles.

   b. Sensor Integration and Data Handling: Simulating how a robot uses sensors to gather data and make decisions based on that data.

   c. Actuator Control and Manipulation: Description: Simulating the control of robotic actuators (e.g., motors, servos) to perform tasks such as picking up objects, rotating, or moving parts.

Answers to Review Questions 5.3: CAD tools

1. The three distinct features of Tinkercad include:
   a. Drag-and-Drop 3D Shapes

   b. Electronic Circuit Design and Simulation

   c. Codeblocks (Block-Based Coding for 3D Models)
2. The primary purpose of the features mentioned include:
   a. Drag-and-Drop 3D Shapes: The drag-and-drop feature allows users to create 3D models by selecting and positioning various predefined geometric shapes (e.g., cubes, cylinders, spheres) in the workspace. Users can scale, rotate, and combine these shapes to build complex 3D models with ease.

   b. Electronic Circuit Design and Simulation: The electronic circuit design and simulation feature allows users to design and simulate electronic circuits. Users can add components like LEDs, resistors, batteries, and microcontrollers (such as Arduino) to their virtual circuits and simulate their functionality in real-time.

   c. Codeblocks (Block-Based Coding for 3D Models): Codeblocks enable users to bring their 3D models to life using block-based coding. Users can define shapes, set their positions, and manipulate them using programming logic.
3. The following procedures apply when preparing a design for 3D printing:
   a. Check Design Specifications: Ensure the model meets all necessary specifications, including dimensions, tolerances, and material properties.

b. Optimize the Design: Adjust the design for 3D printing, considering factors like wall thickness, support structures, and print orientation.

c. Export the Model: Save your design in a 3D printable file format, such as STL or OBJ.

d. Import into Slicing Software: Open the exported file in slicing software like Cura or PrusaSlicer.

e. Configure Print Settings: Adjust settings such as layer height, infill density, print speed, and support structures based on your specific needs.

Answers to Review Questions 5.4: 3D Printing

1. The two 3D slicing software packages:

   a. Ultimate Cura

   b. PrusaSlicer

2. 3D slicing software is essential for 3D printing because it converts CAD models into instructions that 3D printers can understand. It prepares models for printing by defining layers, generating supports, and optimising printing parameters. Additionally, it helps visualise the model, detect errors, and troubleshoot printing problems. By using 3D slicing software, you can ensure that your 3D prints are high-quality and successful.
3. True
4. True
5. b. To convert CAD models into G-codes
6. c. Ultimaker Cura

Extended Reading

1. **For further reading on Programming, Modelling and Simulation Tools**
   a. Installation of Arduino IDE on different operating systems
   https://emanual.robotis.com/docs/en/software/arduino_ide/

   b. Modelling, Simulation, and Control of 4-Wheeled Mobile Robots in ROS and Gazebo - From Scratch! https://www.youtube.com/watch?v=ad2jd8SCK-o

   c. Virtual Robotics Toolkit Tutorials
   https://www.youtube.com/playlist?list=PLD6Y1qUg5LO72JIur1RTdLNX1a5wmmVNO

2. **For further reading on CAD tools**
   a. Tinkercad: Exporting for 3D Printer:
   https://www.youtube.com/watch?v=4FDlgeCOGUI

   b. Getting Started with Fusion 360
   https://www.youtube.com/playlist?list=PLHEouj-sdEJEm_xextUwnfhRJEdUs2hn4

   c. SolidWorks Tutorial for Beginners
   https://www.youtube.com/playlist?list=PLrOFa8sDv6jcp8E3ayUFZ4iNI8uuPjXHe

3. **For further reading on 3D Printing**
   a. Slicing and G Code: The Bridge Between 3D Model and 3D Printer https://support.snapmaker.com/hc/en-us/articles/4409195239575--Slicing-and-G-Code-The-Bridge-Between-3D-Model-and-3D-Printer

   b. Slicer (3D printing): https://en.wikipedia.org/wiki/Slicer_(3D_printing)

ACKNOWLEDGEMENTS

List of Contributors

Name	Institution
Isaac Nzoley	Wesley Girls' High School
Kwame Oteng Gyasi	Kwame Nkrumah University of Science and Technology
Kwame Owusu Opoku	Prempeh College