# **Tangible Code Interacting with Physical Robot**

### Michael Stubbs

Report 2

### Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science Honours in Computer Science and Information Systems in the Faculty of Science at the Nelson Mandela University

### Professor Jean Greyling

### 16 August 2019

# Glossary

API - Application Programming Interface

Constructivism – Educational theory where people learn by taking their pre-existing knowledge and being made to change and improve it.

CS – Computer Science

Digital Divide – The gap between those who have technology and those who do not.

DSR - Design Science Research

SDK - Software Development Kit

Tangible – Something that can be seen, touched or noticed. The opposite of abstract.

Unity – Software development engine.

# Table of Contents

[**Tangible Code Interacting with Physical Robot** 0](#_Toc16608360)

[Michael Stubbs 0](#_Toc16608361)

[Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science Honours in Computer Science and Information Systems in the Faculty of Science at the Nelson Mandela University 0](#_Toc16608362)

[Professor Jean Greyling 0](#_Toc16608363)

[16 August 2019 0](#_Toc16608364)

[Glossary 0](#_Toc16608365)

[Table of Contents 0](#_Toc16608366)

[1. Introduction 1](#_Toc16608367)

[2. Progress Report 1](#_Toc16608368)

[3. Requirements 1](#_Toc16608369)

[3.1 Functional Requirements 2](#_Toc16608370)

[3.1.1 Robot 2](#_Toc16608371)

[3.1.2 Grid and Obstacles 2](#_Toc16608372)

[3.1.3 Mobile application 3](#_Toc16608373)

[3.1.4 Game tokens 3](#_Toc16608374)

[3.2 Non-Functional Requirements 3](#_Toc16608375)

[4. Design 4](#_Toc16608376)

[4.1 Use-Case Diagram 4](#_Toc16608377)

[4.2 Robot Design and Prototype development 5](#_Toc16608378)

[4.2.1 Arduino 5](#_Toc16608379)

[4.2.2 Sphero 6](#_Toc16608380)

[4.2.3 Lego Mindstorms NXT 7](#_Toc16608381)

[4.3 Grid and Obstacle Design 8](#_Toc16608382)

[4.4 Tokens and mobile Application Design 9](#_Toc16608383)

[5. Conclusion 9](#_Toc16608384)

# Introduction

Report 2 contains information on the requirements of the artefact, the design behind it and what tools and technologies exist to aid with the development. The progress and the ideas that took place while considering how to create a physical robot for the children to play and interact with. It compares the progress of the project with that of the Project plan furthermore it discusses the difficulties that have occurred throughout the creation of the artefact.

# Progress Report

|  |  |  |
| --- | --- | --- |
| Design Science Research Task | Project Task | Guidelines |
| 1) Gather information about the research area | 1) Literature Review | 2: Problem Relevance |
| 2) Explain the research and objectives of the project | 2) Define Research Questions | 4: Research Contributions |
| 3) Creating an artefact | 3) Prototype development and design | 1: Design as an Artefact |
| 4) This artefact is reviewed according to evaluation criteria | 4) Testing and analysis | 3: Design Evaluation |
| 5) This artefact is then improved based on these findings and more data is gathered | 5) Improvement of prototype | 5: Research rigor |
| 6) A final conclusion is then reached based off the new data that has been gathered through the use of the artefact | 6a) Final artefact is implemented | 6: Design as a search process |
|  | 6b) Conclusion | 7: Communication of research |

The project structure has been aligned with Design Science Research(DSR). According to the plans and guidelines created in report 1 (Stubbs, 2019) a prototype needs to be designed and evaluated. The project has followed the time-line setup in the initial project plan but some tasks have taken longer than expected. The image recognition has not been implemented yet and the tokens and levels still need to be designed and created. However the correct Bluetooth communication and robot protocols have been setup. The next step in the design and creation process is to create a prototype that can be used to test how children respond to the tangible code and the physical robot and from there improve on this prototype iteratively. Table 1 comes from report 1 and it displays the different tasks the project will take to align itself with a DSR project, this report will include information on the prototype development and design and the testing and analysis of different prototype designs.

Table 1(DSR PRoject alignment)

# Requirements

The requirements for this project are based off Report 1. That report gave details about a proposed solution, details on evaluating that solution and laid out criteria that could be used to compare it with other existing systems. The core requirements for the system can be summarized as:

* It needs to be tangible.
* Affordable
* Mobile
* Convey core concepts
* Be Open Source.

So the functional and non-functional requirements need to properly reflect these core requirements and need to show how the proposed solution will fulfill them.

## 3.1 Functional Requirements

The proposed solution consists of a grid with obstacles, a robot that can receive inputs and move on the grid, levels which will detail how the obstacles need to be placed, and token based instructions that need to be organized in order to complete the level. Some device that will need to read in these tokens and send commands to the robot and then learners who are playing with the system. All these separate components working cohesively together is a requirement of this system.

### 3.1.1 Robot

The robot needs to be a physical actor that accurately represents the learners attempt at the level. It has to receive input from the user and then act and if it encounters a collision it needs to send this information back to the device sending it instructions. One of the requirements of this project are for it to be mobile so the robot needs to be able to communicate with a mobile device both to send and receive commands. Two common communication protocols for this are Wi-Fi and Bluetooth. Bluetooth being preferable as it is easier to setup a connection between the two devices and some places where the robot will be used might not have Wi-Fi or the robot might not be able to connect reliably. The robot needs to be able to move accurately, as if it is unable to do so it might not move to the intended position on the grid. If it is moving to unattended positions due to being imprecise it will ruin the point of the game and disrupt the learning activity. The accuracy of the movement can be determined by how consistent its forwards and backwards movement are as we need to make sure it moves into the intended block on the grid correctly. How accurately the robot turns is important as if it is off by even a few degrees these errors will compound into the robot facing an unintended direction. It needs to have accurate sensors to detect collisions as that is how it will interact with the obstacles. It has to be durable as there will be children playing with it, this means that anything that can easily torn apart or is difficult to reassemble should be discouraged. The robot needs to be relatively simple to setup as if it is overcomplicated producing more systems would rely on someone with a high technical skill.

### 3.1.2 Grid and Obstacles

The playable grid needs to be an accurate way of tracking the robot throughout the game’s lifecycle. Each block on the grid needs to be able to fit the robot in it with extra space to accommodate small errors and leave it enough space to turn inside of a single block. It cannot be too big as it needs to be able to be transported and lay flat inside of classrooms. The grid is important as having a flat surface which has a consistent material is important to ensure the accuracy of the robot.

The obstacles are simply objects that will be placed on the grid that need to stop the robot from moving through them. This will force the learners to turn and avoid them. They need to be big enough to see and easy to place.

### 3.1.3 Mobile application

The mobile application will be what enables the learners to play and it will have 3 main roles:

* It will be a server which the robot connects to and it will handle the connection between these 2 devices as a server.
* It will be used to scan the game tokens and handle sending them to the robot.
* It will keep track of their progress and statistics while playing.
* It will allow the users to select a level they are on and how to setup the grid.
* It will give tutorials and help to explain to the users what is happening.
* It will give feedback to the users so they can see what skills they have learnt and what occurred during their playtime.

### 3.1.4 Game tokens

The game tokens are what the learners use to tell the robot how they want it to move. They therefore have to be human readable and have to be easy to interpret. They have to convey simple programming concepts like IF and While statements and when they are placed together they need to display sequential logic to the learners. They should be physical in nature and should easy to understand ensuring the constructivist learning approach. They need to have a way of being inter-connected such that the mobile device can interpret and tell the robot what it must do. They have to be able to be picked up accurately through the mobile devices camera.

## 3.2 Non-Functional Requirements

The non-functional requirements for a system are those that are not linked to any one component and that can be used to judge the system as a whole. The non-functional requirements for this system are as follows:

* The cost involved - as stated in report 1 cost constitutes a large problem for learners who do not have enough resources to learn about the IT field.
* Durability of physical components– The solution needs to be semi-durable as children will be playing with it. If they break it while playing and it has to be rebuilt or repurchased there will be disruptions in the learning function.
* Open source – The project and system being open source are important, as it allows additional code for the system to be developed, or adapted based on the situation at hand. Also if the more advanced learners become more interested in the code and want to learn and build applications themselves it is important to enable this.
* Accessibility – South Africa is a diverse country with many different languages and cultures. It is important to develop with respect to this and make a system that all the different children from various backgrounds can understand.
* Learnability – The main focus of this system is the education of children, so they need to be able to learn the system which will then encourage them to learn about the IT field. The system should be easy to learn so they can understand the game they are playing correctly and so when the game gets harder and more complex they are still not stuck on the basics.
* Response time – The system cannot take too much time to accomplish its goal. The mobile device should read the commands in and then almost immediately the robot should start moving. This is important as you want to keep the learners attention and let them see the outcomes of their actions as swiftly as possible.
* Usability – The systems user interface needs to be properly designed. As we want it to be intuitive for the users some of which will have very little experience with computers. The more usable the interface, the faster the learners can learn and less user errors will occur.
* Safety – Because the system is being designed for children we have to specifically take their safety into consideration. Meaning for things like our robot we should ensure that they cannot hurt themselves on it while they play.

# Relevant Technologies

This section will look at what technologies have been relevant to this system throughout the design phase and will look at providing insight into how the technologies work and what they will be used for.

### 4.1 Bluetooth

Bluetooth is relevant to this project as it is what the system will use to connect to the robot and stream data between the two devices. There have been two different types of Bluetooth investigated in this project and that is Bluetooth Classic and Bluetooth Low-Energy. Bluetooth classic devices work by taking one device and making it discoverable and then the other searches/scans for it and finds it and attempts to pair. If the devices have been connected before and are still on each other’s device list they can simply pair but if they are not a pin needs to be shared between the two devices in-order for them to trust each other so they can pair. When they have been paired one device acts as the server and manages the connection while the other is a client. The two devices can share and exchange data streams until the connection is closed (Usability Expert Group , 2017). Bluetooth Low Energy (BLE) was the other type of Bluetooth investigated. In classic Bluetooth the two devices need to be paired and share a constant stream of data until the connection is closed however with BLE devices the devices do not need to be paired to share data. With BLE devices instead you can just share data one-way and you do not need to be connected prior to doing this. This allows for multiple devices to be stream data at once and it saves power on those devices. The Bluetooth classic pairing scheme is much more power hungry as the devices need to constantly be connected to each other. BLE devices scan based on proximity and find which devices are in the area that they can transmit to, and then pick the correct to stream the data to (Developer.Android, 2019).

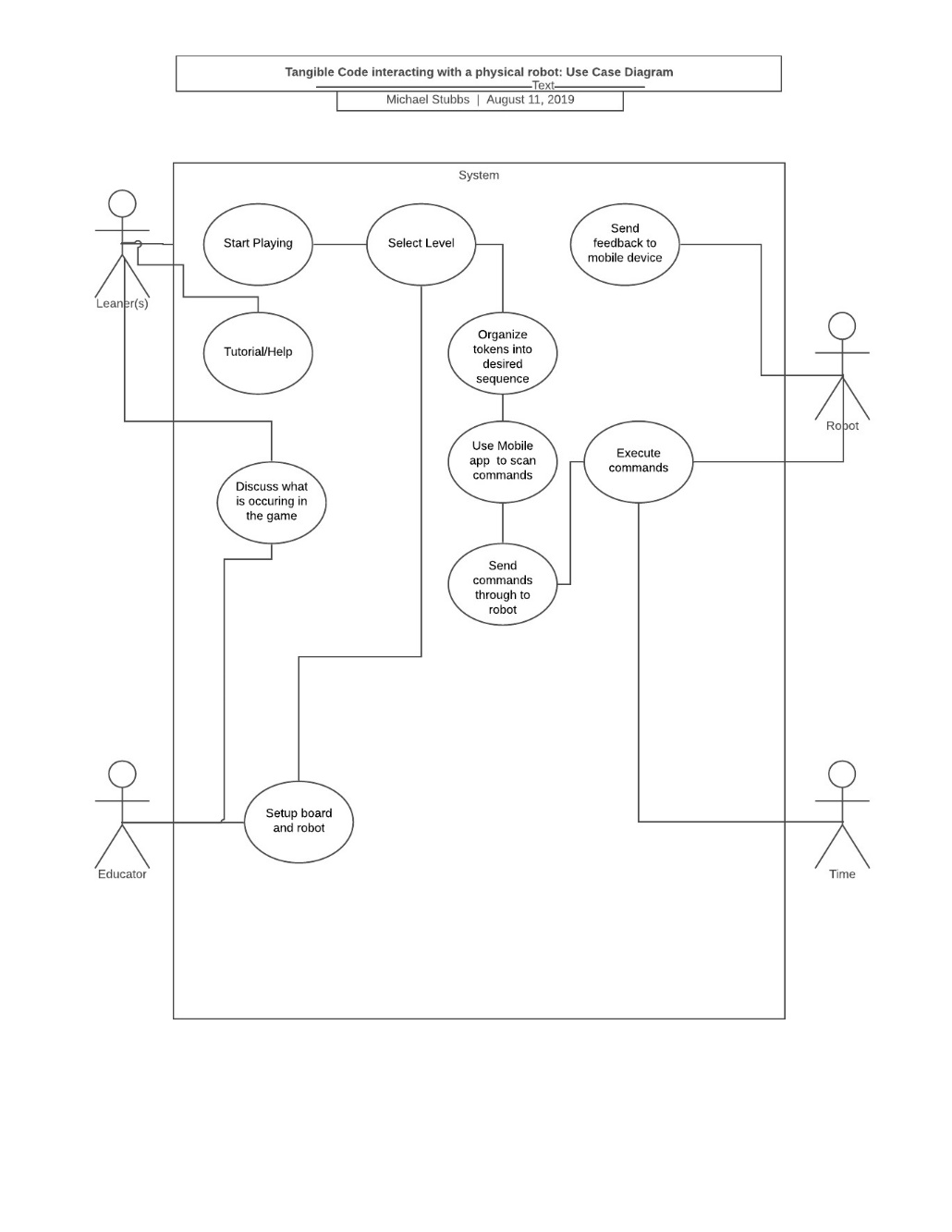
* 1. SDK’s and API’s

# 5. Design

Chapter 3 defined both the functional and non-functional requirements of this project. The design phase will focus on a system that fulfills these various requirements. Design in Design Science Research is an iterative process where prototypes are created and then iteratively improved on. This means that building prototypes is part of the design phase and so that phase will be included as a section of this chapter. A Use-Case diagram will be given in section 4.1 and this be used as a basic guideline of the actions that the system has to accomplish.

## 5.1 Use-Case Diagram

A use case diagram is an effective way of representing a system and the various actors involved. This diagram shows how one or many learners could be playing the game and how the educator and they are in-charge of setting up the game state. It shows the basic game sequence of selecting a level, organizing the tokens into the desired sequence, scanning the tokens and then the robot moving on the grid. The learners will then either pass or fail the level and will have a chance to talk to their peers and educator before moving on or trying again.



## 5.2 Robot Design and Prototype development

The design and choosing of the correct robot for this project was challenging and there lots of mistakes made throughout this process. Luckily DSR supports these mistakes and sees them as a cost of doing research. This does mean that this phase took a lot longer than originally quoted on the project time-line and it is the reason this project is behind its predicted schedule. That being said the current iteration of the robotic prototype fulfills all the requirements laid out for it and the process of getting it there will be discussed in this section. There were three different robot systems looked at in-depth for this project and two of them were prototyped. Sub-Section 4.2.1 will discuss the Arduino board and why it was considered for this project. Section 4.2.2 will discuss the Sphero Mini and that prototype. Section 4.2.3 will discuss the prototype robot for the Lego Mindstorms NXT and justify why that is the robot being used.

### 5.2.1 Arduino

Arduino is an open-sourced hardware and software company that makes micro-controllers for both industry and personal use (SparkFun, 2019). They are highly configurable and have been used for a large amount of projects. They make a variety of different boards but the most beginner-friendly board is the Arduino Uno R3. The robot designed consisted of 2 motors, 2 wheels, a motor driver, an ultrasonic distance sensor, battery pack and a Bluetooth module. The robotic car that was designed would have been made based on the designs from a company called DroneBot workshop (Workshop, 2017). All these parts and the additional wires and tools to build an Arduino would come to around R1000 on a South African online store called HobbeyTronics (HobbyTronics, 2019). The Arduino fulfills many of the requirements as it is cost effective, is open sourced, it has Bluetooth communications, has a sensor to detect collisions.

However the setup of an Arduino requires knowledge of electrical circuits and the ability to weld wires correctly. Too make a motor move you have to send voltage to that specific motor, this is an issue because it becomes difficult to measure rotations and get accurate movement. It would require considerable time to learn about how these circuits work, the programming language for Arduino’s and to get the accuracy that is required for this project would be out of scope. The difficulty to build is also an issue because it requires skill to setup/repair and it is difficult to find people who are knowledgeable with an Arduino to do so. It breaking is an issue because it would be small and fragile and if a kid stepped on it or was too rough with it, it might break. Also there are wires that can be tugged and it poses a safety hazard for the children. It was decided that an Arduino based system was not the correct robot for this system due to how fragile it is and how difficult the robot is to setup in comparison to other robots.

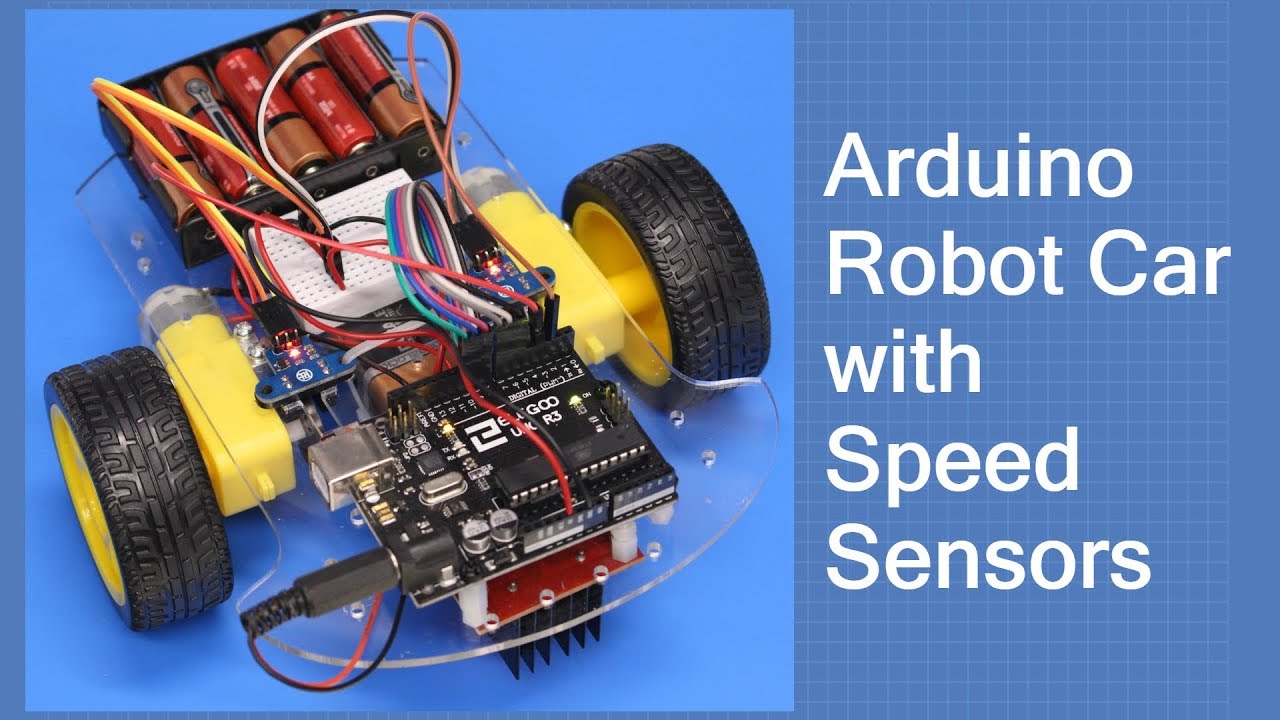


Figure 1 Arduino robot car (Dronebot workshop,2017)

### 5.2.2 Sphero

Sphero is a company that makes a wide range of programmable spherical robots with a focus on educating children about STEM topics. They have an IOS, Android and desktop app that can be used to drive, program the robot with programmable blocks or code for it using JavaScript. Sphero made their first robot in 2011 and currently have 3 different educational robots on the market. The cheapest of the 3 is called the Sphero Mini and one was purchased for this project. It is a robust sealed sphere-shaped unit that can be run and setup by simply plugging it into a computer (Sphero, 2019). The new Sphero’s are all Bluetooth Low Energy Devices.

When investigated it looked as if it had an open source SDK for Android, IOS, JavaScript and Unity (sdk.Sphero, 2019). However these libraries have not been maintained and are no-longer Open Source. The Unity library was the first software that was tested as Unity allows for the creation of cross-platform games and has lots of resources to aid in developing apps and games. The Unity library was last updated in 2013 (Unity-AssetStore, 2013) and has not been updated for the modern versions of Unity, Android devices or modern Sphero’s. There was an attempt made to update their library for the latest versions of Unity but this did not work due to some of their assets being hardcoded to work with older versions. Next an older version of Unity was used, one that worked with this plugin. The issue with this was that the Bluetooth on Android devices and their Sphero’s have changed but those versions of Unity and their plugin do not have support for that. The next attempt was to try build an app for it inside of Android-Studio using the SDK available from their Github (Github/Sphero-inc, 2016). Their SDK is no longer Open-Source and has not been updated since 2016. This is an issue as the way Android handles BLE connections has changed and their library has not been updated to reflect this change. So it is not possible to scan for the Sphero-mini using the SDK provided on mobile phones running newer versions of Android and it is not possible to update the SDK as it is not Open-Source. Without the use of the SDK an App that handles the connection between the Sphero-mini and mobile device is out of scope. An attempt was made to try find a more recent Android API or SDK for the Sphero-mini from the developers and engineers at Sphero but the feedback received was that they were still working on it and were not sure when an updated one would become available.

The Sphero-mini could be programmed through JavaScript, so a JavaScript server was hosted on the phone by making use of Node.js. The robot managed to successfully connect to the Node.js server but there were other issues which arose from this, as every single time the application was run the Node.js server would have to be started on the cellphone and that also required some technical knowledge and setup. During this testing other issues became apparent as well. The Sphero mini’s collision sensors were not accurate and neither were its turning and movement completely accurate. These are important requirements for the robot.



Figure 2 Sphero Mini (Takealot, 2019)

### 5.2.3 Lego Mindstorms NXT

Lego Mindstorms is a house-hold name when it comes to teaching children about robotics and code. As discussed in report 1 they created some of the first educational robots and were the first to make a system that made use of code blocks to control a robot. The NXT is an older model of the Lego Mindstorms franchise but for this system it still fits perfectly. It has various different types of configurations and has various hardware modules that can be removed or added. It has Bluetooth and Infrared communication. By making use of Lego it makes the robot seem more approachable to children, and it is also fairly durable but importantly it is easy to fix. The design and creation of the robot follow a normal Lego instruction book, meaning that if does break or is taken apart rebuilding it is as easy as building Lego.

Lego Mindstorms boasts Open-Sourced software and has an openly available API that many developers have made use of to create their own different Operating systems that can be run on the Lego Control brick. The default operating system allows for the creation of programs on a Computer or mobile that are then run on the brick but because there is the ability to flash it with a new OS it can be altered to fit the specific needs of a project. This is important as this project makes use of one of the operating systems called Lejos NXJ (Lejos, 2019)

Lejos is a project that started in 2005 and it is an operating system that runs a scaled-down java virtual machine which essentially lets it run java. The Lejos organization has made operating systems and libraries for both the NXT and the new EV3 bricks and there are still active members developing for both of them. Java is an Object Orientated language and by using it a better program for the Lego NXT brick can be developed, by making use of key functions like Java Threads, synchronization and object types. The use of Java makes handling the communication between the robot and mobile device simple and it gives a way of properly tracking movement and collisions. The communication is done by creating a Bluetooth server on the mobile device which the robot then connects to. Once it has been connected an Input and Output thread are created on both sides of the communication and this is then used in conjunction with a robotic protocol that was developed to stream data between the two devices. It has accurate movement sensors, which track and regulate the motors by making use of a tachometer. A tachometer is a device that can detect the rotations of a motor, this means that by using tacho counts we can accurately control the rotations of the motors. The code to use the motors in Java is relatively simple and it allows for the creation of a fast and accurate system for tracking and controlling movement.

Various configurations and setups of the NXT were explored as many different configurations can be made and it is really up to the designer to create one that best suits the system. A Tank shape and tracks were chosen because of how a tank can turn on the spot by rotating one motor forward and one backward versus a normal 2 wheel drive car which requires an arc to turn. This robot does the best at satisfying as many requirements as possible but it is difficult to acquire currently and can be costly. The robot designed takes up a space of 17,5x 13,5 cm.



Figure 3 Lego NXT Tank

# 6.Future Designs

## 6.1 Grid and Obstacle Design

The grid could only start to be designed when the robot design and shapes were finalized. This occurred because the sizes of different robots vary so greatly and also the surfaces and space they need to operate vary. Because the current robot can turn on the spot the grid needed to be able to support this. So an 8x8 grid with Dimensions 25x25 cm per square was designed. This design allows for the robot to have enough space to turn and travel. If the grid is too big there is an issue of how long it will take the robot to move between squares and thus making an attempt at the game take longer than necessary. There is still an investigation going on trying to determine what materials to use as things like cost, portability and stability all need to be taken into consideration.

The obstacles need to be able to be clearly picked up by the Lego robots ultrasonic sensor. What seems to have worked best for this is larger objects that can be detected from 360 degrees by the robot. So currently cubes that fill one singular square in the grid are being designed and will be tested with the grid.

## 6.2 Tokens and mobile Application Design

It was decided for this project it might be best if a current mobile game that already makes use of tangible coding tokens was ported to work with the robot. Due to how long the robot setup took and how it would be better to make use of product that has already been tested and evaluated. By making use of a system that has already been created the use of a robot can directly be compared to the use of just the mobile game. There will be further interface designs, token designs and level designs done when the correct application to port the robot to has been chosen.

# 7 Conclusion

This report details the process of defining the requirements for the project. Looking at what obstacles there are when dealing with children and what obstacles have occurred through the iterative development and design process of DSR. The designs for this project have been done by taking into careful consideration what was laid out during the requirements for this project. Each different element from the Robot to the Grid to the Mobile App have all been done by taking into consideration what is occurring during the current iteration of the project.