

Final submission - 1902008

by Abigail Crimlis

FILE	17513_ABIGAIL_CRIMLIS_FINAL_SUBMISSION_-_1902008_241917_1139973135.DOCX (952.86K)		
TIME SUBMITTED	07-MAY-2020 11:26AM (UTC+0100)	WORD COUNT	2769
SUBMISSION ID	126575920	CHARACTER COUNT	13213

Integrating project

Individual assignment

Module code: M4X01506

Student no: 1902008

Student name: Abigail Louise Crimlis

1. Introduction PG 1
2. Environmental factors PG 1
3. Movement and motors PG 2
4. Programming PG 2 -3
5. Break down of the problem PG 3 - 10
 - 5.1 The coloured line PG 4 - 7
 - 5.1.1 Issues surrounding each mode
 - 5.1.2 Staying on track
 - 5.2 Traversing the Block PG 7 - 9
 - 5.3 Negotiating the maze PG 9 - 10
6. Logical Solution PG 11 - 14
 - 6.1 Flowchart the Line PG 12
 - 6.2 Flowchart the Block PG 13
 - 6.3 Flowchart the Maze PG 14
7. Summary PG 15
8. References PG 16

PG 1

1. Introduction

The problems set out before us can be broken down into three separate stages, these are as follows:

The robot must follow two coloured lines taped to the floor, they will be of the colour red and blue. We do not know how long the line will be or the shape in which it will follow but we do know that at some point the line will loop and then continue forward.

Placed on the line will be a blue wooden block, the robot will need to traverse this block in some manner and then continue following the coloured line until it has reached the end point. This will be noticeable via a green patch of tape.

The next and final stage of the mission will be a maze. Again, the layout of the maze is unknown, the robot must be able to deal with any type of layout it may incur. The objective is to reach a red block in which the robot will then know it has succeeded, the robot will then make a sound once it has completed the mission. The maze will be made of cardboard, if the robot touches the cardboard causing it to break, the mission has failed.

2. Environmental factors

The environment in which the robot will be conducting the test can have a major impact on the way in which the robot can perform.

Listed below are the environmental factors that need to be incorporated within the design.

- Flooring – The challenges will be based on a cream carpet. Carpet offers better traction meaning the robot will be less likely to fall over, although turning is less accurate.
- Lighting – as discussed in the 'breaking down the coloured line' section ambient light can affect the different sensors to be used.
- Open space – The challenge will take place in a computer lab, there may be objects in the way or space constricts. The robot should be able to navigate any unaccounted-for objects by using the sensors it has equipped.

3. Motors and movement.

PG 2

The ev3 Mindstorms kit contains 4 motors. The large motors whose main purpose is to rotate the wheels. The second type that comes with the box is a medium motor, primarily used for grabbing mechanisms and forklifts etc.

Large servo motor: Can run at 160 – 170 rpm with 1 degree of accuracy. The torque can run up to 20ncm, meaning it is relatively fast. The motors can be attached to an axis on the base of the robot and then connected to the provided wheels. The motors will run in sync together using the pre-defined programming block in the Mindstorms education centre. [4]

Medium servo motor: Not as powerful as the larger motor, running at speeds of 240 – 250 rpm with 8ncm of torque. The smaller motor is faster but would not be strong enough to power the wheels and is better suited for moveable components on the robot. [4]

Another option would be the track that is included within the box, the track can create vehicles such as a tank or a production line carousel. The track would need to be connected to a rotational design, implemented with one of the motors discussed above.

For the robot to gain better traction and more control on turns, 4 large motors will be used to connect to the wheels.

4. Programming

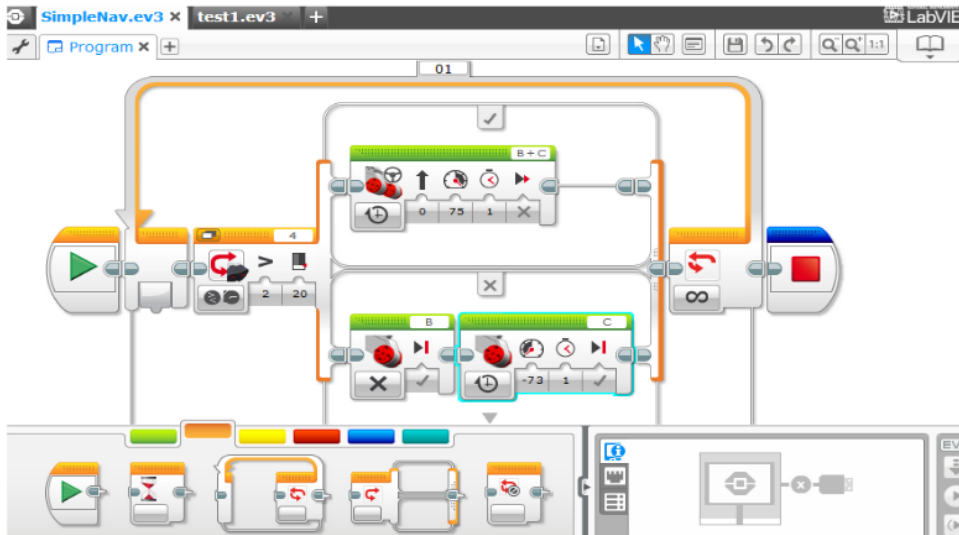
The Lego Mindstorms kit includes the EV3 Intelligent block, a programmable brick that can be charged through USB or through Batteries. The ARM9 Processing unit is at the core, with Wi-Fi adaptability as well as Bluetooth capabilities. The users can add 4 motors and 4 sensors at any one time on to the EV3 brick. There are two options available for programming the robot:

- Lego Mindstorms education software

Adopting a “drag and drop” programming method. Users are able to learn the concepts and basic principles of programming without the need of learning the syntax. The Mindstorms software has built in functions that communicate with the sensors and motors connected to

PG 3

the robot. It also includes programming paradigm such as IF/ELSE statements, for loops, while loops and switch statements.



Src: lego.com

- MicroPython

On the body of the Intelligent brick is a SD card slot. This can be used to download the python programming language. Like any programming language microPython can be written on a text editor such as Visual studio code. microPython is a small subset of the Python 3 language. Equipped with its own compiler and running resources, it was developed to be used on micro controllers and in constraint environments. It also comes with predefined connections for the EV3 sensors. [5]

```
testing_motor = Motor(Port.B) – storing connection to motor in port B.  
testing_motor.run_target(500,90) – Running motor up to 500 degrees per  
second until 90 degrees is reached.
```

5. Break down of the problem

To gain a better understanding of the problems and to find the best solution, each problem needs to be broken down and possible solutions

to be discussed to find logical errors and find the best resolution to achieving our goals.

5.1 The coloured line

To be able to follow a coloured line, the robot needs to be engineered with this in mind. The ev3 mind storm contains sensors that makes this possible, the colour sensor has the ability to emit white light as well as RGB colours and return the information to the colour sensor. There are 3 different modes available:

- **Reflected light intensity mode:**

Emitting a red light on to the surface that is to be tested, the reflected light is transferred back into the colour sensor. The returned value can be anywhere from 0 (very dark) and 100 (very light)

- **Ambient light intensity mode:**

Like the Reflected light mode, the ambient setting is used to test the amount of light within the environment.

- **Colour mode:**

src = lego.com



There are seven different colours the EV3 sensor can detect.

• Yellow (4)	• Red (5)	• Black (1)
• Brown (7)	0	7
• Blue (2)	• Green (3)	• White (6)

They are outputted in the form of an integer value between 0 and 7.

5.1.1 Issues surrounding each mode

To best design the solution a mode needs to be decided as this will impact on where the sensor needs to be placed to get the best results.

Ambient and reflected intensity mode does not suit our needs in this instance. These modes are better suited for following a torch or for a robot that should only move during the day.

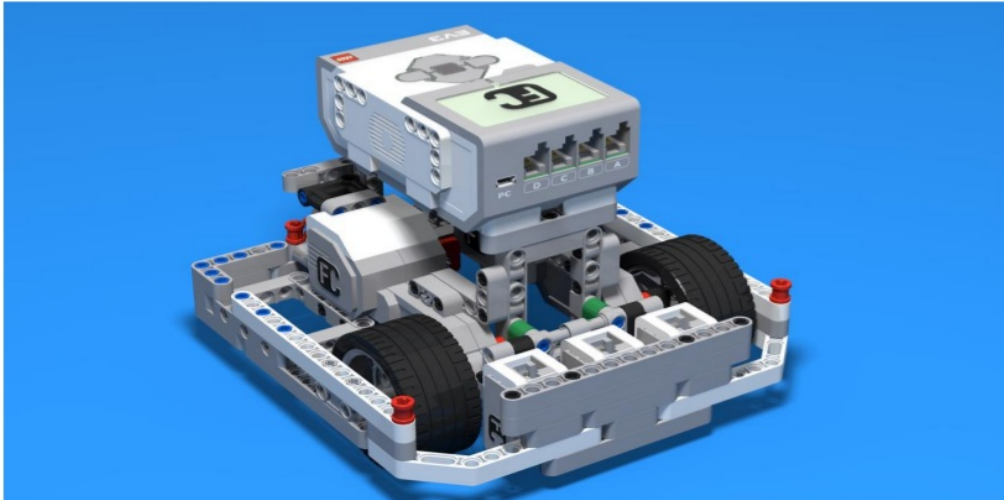
The colour mode has more properties that will be useful as it can detect the colour of the lines, (red and blue), according to *Burfoot. J* for the sensor to work correctly the sensor must be placed 1-2cm away from the testing surface and have consistent lighting. [1]

The amount of ambient light in the room can largely affect the value returned as light changes the way in which colours are perceived.

There are two possible solutions for combatting this issue.

1. Avoid shadowed areas, close all blinds and use artificial light within the room.
Although a simpler solution, there would not be a guarantee of accuracy, making the robot less efficient and unusable for similar tasks it may face in the future.
2. Create a box/container around the robot to limit the light intake and keep a consistent ambiance surrounding the sensor.

From an engineering perspective it may not be the easiest thing to accomplish but would mean the robot is more versatile when it comes to performing in different environments.



Src = Fllcasts.com

5.1.2 Staying on track

To ensure the Colour sensor can receive the correct colour at all times it needs to be programmed in a way that will allow this to happen. There are a few things that need to be considered.

1. The robot needs to be able to see the blue line at all times and automatically correct itself when it has strayed.
We can accomplish this by programming the robot to turn 1 degree to the right if the sensor detects the red line.
2. The robot needs to be slow enough to give the sensor time to analyse the sensory input.
This can be done by adding the 'wait' timing function with the parameters of 2 seconds and then moving by half a wheel rotation.

To summarize this section:

The robot will need to be designed in a manner that will encompass the colour sensor to limit the ambient light reaching the sensor.

The mode will be set to "Colour Mode" and needs to output the integer 2 (blue) to be able to continue forward. If the robot detects the integer value of 5 (red) it should turn right by 2 degrees.

The robot needs to wait for 2 seconds to ensure the colour sensor has time to process the input values. It will then move forward slowly in order to keep the robot straight.

5.2 Traversing the block

During the line following section of the mission, a blue wooden block will be placed on the line blocking the robot from continuing its initial task. There are two sensors included within the EV3 kit that could be a possibility for this part of the mission.

Ultrasonic sensor:

The ultrasonic sensor emits soundwaves to detect and measure the distance between objects. The sensor can detect objects within a range of 1cm – 250cm with an accuracy of +/-1cm. [2]



Src: lego.com

Touch sensor:

The analog pressing mechanism can detect when the button is pressed and released with much accuracy. It can also count single and multiple presses in rapid succession.



Src: lego.com

There are a few methods that we can use to complete this task.

1. Move around the block

Using the ultrasonic sensor, we could detect when the robot is within 20cm of the box. Once the box is within distance the robot should turn left, move forward, turn right, move forward and then right again, moving forward until the blue line is detected. The robot will then need to turn left to straighten up and continue on the colour line.

1.1 Problems that can occur

Hard coding the robot's movements makes this not the most efficient way, as it will be unreliable in future cases that require navigating around an object. To combat this issue another ultrasonic sensor could be placed on the right of the robot, following the lines of the block, meaning the robot could turn when the sensor on the right no longer detects an object.

2. Climb over the block.

Using the ultrasonic sensor, we can detect when the robot approaches the object. When the robot is within 10cm, extendable legs would extend and create a stepping motion over the block.

2.1 Problems that can occur

Engineering mechanical legs that can extend and recoil into itself would be difficult to create using the available materials given in the EV3 kit. It would also not be practical for the remaining parts of the challenge.

3. Sweep the block.

Once the touch sensor has been activated by the robot pressing on to the block, a mechanical arm could be used to swipe the block out of the way. The robot would also need to turn with the movement so that it does not render itself broken.

3.1 Problems that can occur

Again, this would be quite difficult to engineer, as the weight of the block is unknown, it would be hard to compensate for the force needed to move the block out of the way without flipping/breaking the robot. It would also be impractical for the maze.

4. Pick up the block and move out of the way

Using two mechanical arms on both sides of the robot, once the ultrasonic sensor detects the block is within 10cm of the robot's position, the arms can lower and grab the brick. To know that the block has successfully been grabbed, touch sensors can be placed on both arms. The robot would then lift the block, turn left 90 degrees, lower the arms, and release the grip. The arms would then need to levitate again and the robot to turn 90 degrees right to continue the line.

4.1 Problems that can occur

As the length and height of the block is unknown there is a possibility the Lego arms would fall short of the length needed in which to initiate its program, the block may also be too heavy for the arms to deal with, or too wide for the arms to grab successfully.

Summarize this section

The robot will move around the block, using two ultrasonic sensors.

Two ultrasonic sensors will be used for navigation, one on the right of the robot and another on the front, this will remove the hardcoded turns.



5.3 Negotiating the maze

The final stage within the assignment; to navigate around the maze the robot will need to be equipped with an ultrasonic sensor. As discussed in the opening section of this report, we do not know the size or shape of the maze. We do know that it will be made of cardboard.

1. The right-hand side rule.

The algorithm is commonly used to resolve the issues faced when programming a maze function. The algorithm dictates that the robot should always turn right and that it will always find the exit.

When the ultrasonic sensor alerts the wall is within 10cm, the robot will turn right, pause and scan, continuing straight if able or turning right 90-degrees until it can move forward. [3]

2. Speed

For the robot to successfully navigate the maze without causing destruction, it is important that it moves slowly. Giving the sensors a more accurate reading.

When the red block is detected, the robot must alert that it has successfully completed the assignment. EV3 Mindstorms has a 'message' option within the programming software, that comes equipped with many different sound files.

To summarize this section

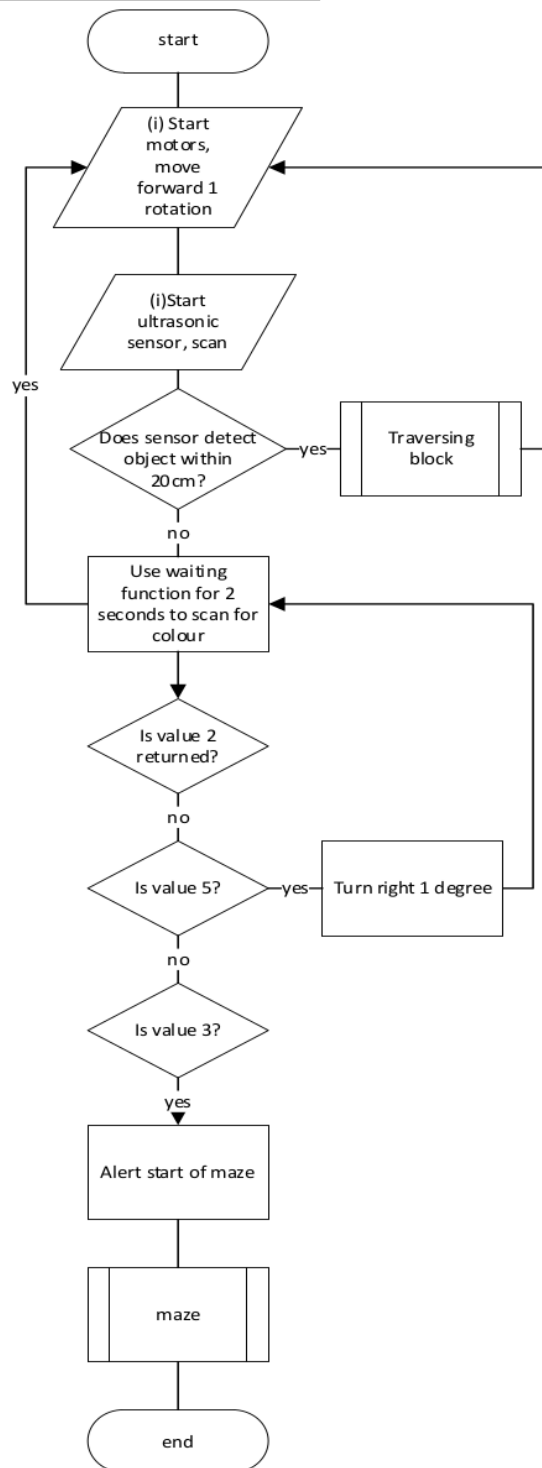
The robot will be equipped with an ultrasonic sensor, it will alert the robot when an object is within 10cm.

The robot will then turn 90 degrees and scan again. Once the robot is no longer obstructed it should move forward at a slow speed until the red block is seen by the colour sensor.

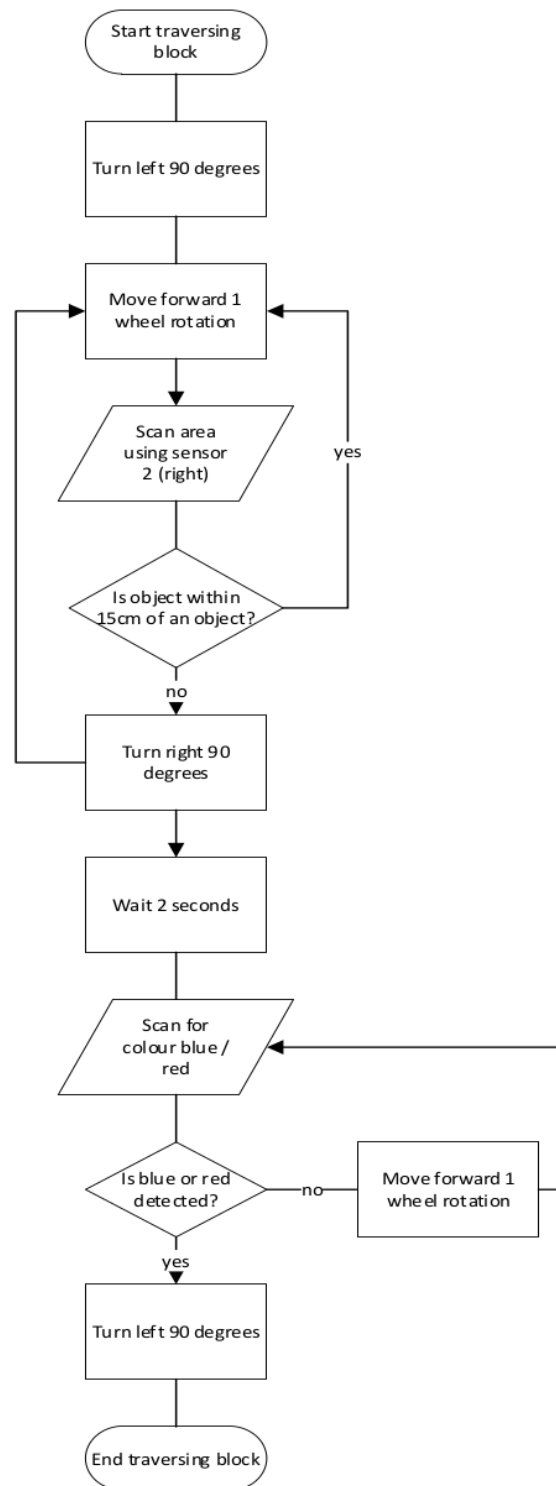
An alert message will then be initialized to signal the end of the task.

6. Logical solutions

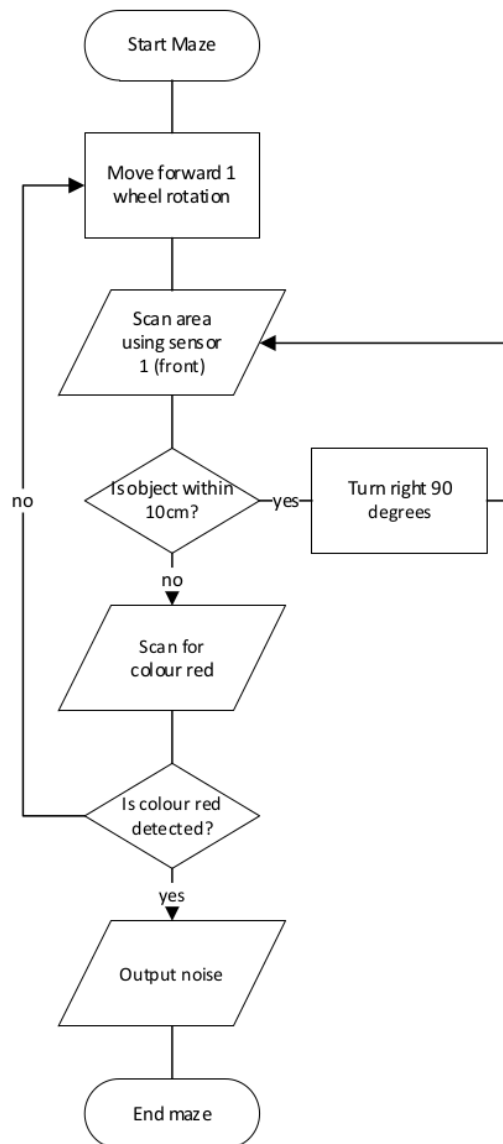
6.1 Flowchart the coloured line



6.2 Flowchart traversing the brick



6.3 Flowchart maze



7. Summary

To summarize:

- If the colour sensor detects the blue line it will continue moving forward, if the red line is detected the robot will correct itself by 1 degree until it finds the blue line again.
- The robot will move slowly to give the sensors a more accurate reading.
- The block will be traversed by using two sensors to navigate around the block, turning at 90-degree angle when the sensors no longer detects an object.
- The maze will be navigated using the right-hand side method; If an object is detected the robot will turn right 90-degrees until it is no longer obstructed.

From a design point of view:

- The robot will be equipped with the following sensors:
 - 1x colour sensor
 - 2x ultrasonic sensor
 - 4 x large motors

The colour sensor will be encased within a block to overcome the environmental issue of ambient light.

The 2 ultrasonic sensors will be placed at the front and right side of the robot respectively.

The robot will be programmed using the Mindstorms education sensor.

References

- [1] Available online: <http://www.legoengineering.com/ev3-sensors/> John Burfoot, 22nd March 2018. Accessed: 15/04/2020
- [2] Available online: <https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-ultrasonic-sensor/45504> - Accessed: 20th April 2020
- [3] Available online: <https://magbuhatrobotics.files.wordpress.com/2017/05/aol-chapter-7.pdf> - Accessed: 24th April 2020
- [4] EV3 Technologies: Available online: <https://counties.agrilife.org/gillespie/files/2015/04/EV3-Motors-Sensors-Explained.pdf> Accessed: 24th April 2020
- [5] Lego education: "Getting started with EV3 MicroPython" Version: 1.0.0. Published online: 26th March 2019. Available online: <https://le-www-live-s.legocdn.com/sc/media/files/ev3-micropython/ev3micropythonv100-71d3f28c59a1e766e92a59ff8500818e.pdf> Accessed: 27th April 2020

FINAL GRADE

74/100

GENERAL COMMENTS

Instructor

Your enthusiasm and ability in design and programming are great to see and you have done a good job of adapting the assignment to the circumstances.

Project Introduction 10% 7

The Introduction is well structured and shows a good logical exploration of the problem. You have a good and succinct writing style. Sometimes in academic writing it's good to add explanation, justification, what if.... Your style is ideal for a technical report however.

Line Follower 25% 20

A clear analysis of the problem with drawings and code modules to support your designs.

Block avoidance 25% 20

Again clear explanation and analysis with examples provided. Good consideration of the practical issues anticipated.

Maze 25% 20

Good analysis which is supplemented by demonstrations in class before lockdown.

Overall summary 15% 11

A bullet point outline of what has been achieved. It's fine to write in bullet points when you have a genuine list like this, writing in paragraphs is often required in academic English.

Assessment Component 2 – Coursework - 40% 76

Flowcharts are presented with a good design solution, these were backed up by early demos in class with different solution attempted.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17
