GNG1103

Design Project User and Product Manual

Red Light Dead Light

Submitted by:

THE MACHINES

Eli Grant

Steve Biju Thomas

Émile Veillette

Evan Huang

December 3rd, 2024

University of Ottawa

Table of Contents

Table of Contents ii

List of Figures v

List of Tables vi

List of Acronyms and Glossary vii

1 Introduction 1

2 Overview 2

2.1 Cautions & Warnings 3

3 Getting Started 4

3.1 Configuration Considerations 7

3.2 User Access Considerations 7

3.3 Accessing/setting up the System 7

3.4 System Organization & Navigation 8

3.4.1 Physical Prototype 8

3.4.2 Software Prototype 8

3.5 Exiting the System 9

4 Using the System 10

4.1 Setup Function 10

4.1.1 Origin Sub-Function 11

4.2 Round 1 Function 11

4.2.1 Scan Sub-Function 12

4.2.2 Return Home Sub-Function 12
4.3 Rounds 2-5 Function 13
4.3.1 Target Sub-Function 13
4.3.2 Player Sub-Functions 14
4.3.3 Worst Case Sub-Function 15
5 Troubleshooting & Support 16
5.1 Error Messages or Behaviors 16
5.2 Special Considerations 16
5.3 Maintenance 16
5.4 Support 16
5.4.1 General Support 16
5.4.2 Code-Specific Support 17
5.4.3 Robomaster S1 Specific Support 17
6 Product Documentation 18
6.1 Subsystem 1: Turret Movement 18
6.1.1 Turret BOM 18
6.1.2 Equipment List 18
6.1.3 Instructions 19
6.2 Subsystem 2: Player Recognition and Tracking 19
6.2.1 Player BOM 19
6.2.2 Equipment List 20
6.2.3 Instructions 20

6.3 Testing & Validation 20
6.3.1 Player Timing 21
6.3.2 R1 Vision Marker Recognition 21
7 Conclusions and Recommendations for Future Work 22
8 Bibliography 24
APPENDICES 25
9 APPENDIX I: Design Files 25
10 APPENDIX II: Other Appendices 26

List of Figures

Fig. 1: 2

Fig. 2: 4

Fig. 3: 5

Fig. 4: 5

Fig. 5: 6

Fig. 6: 6

Fig. 7: 7

Fig. 8: 10

Fig. 9: 11

Fig. 10: 11

Fig. 11: 12

Fig. 12: 12

Fig. 13: 13

Fig. 14: 14

Fig. 15: 14

Fig. 16: 15

List of Tables

Table 1. Acronyms vii

Table 2. Glossary vii

Table 3. Turret BOM 18

Table 4. Player BOM 19

Table 5. Referenced Documents 25

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
DJI	Da-Jiang Innovations Science and Technology Co., Ltd
R1	Robomaster S1
RLDL	Red Light Dead Light
UPM	User and Product Manual
LAWS	Lethal Autonomous Weapon Systems
MAC	Mines Action Canada
QOL	Quality Of Life
BOM	Bill of Materials

Table 2. Glossary

Term	Acronym	Definition
Sub Function	SF	Specific task within a function/objective
Variable	N/A	A value that is capable of change
DJI vision marker	N/A	Refers to physical symbols that the R1 can track/detect
Digital Dehumanization	N/A	The act of converting meaningful human decisions into data, and its consequences

1 Introduction

This User and Product Manual (UPM) provides the information necessary for game hosts and Mines Action Canada representatives to effectively use the Robomaster S1 (R1) to host the Red-Light Dead Light (RLDL) game. This document also goes through previous prototypes, maintenance/troubleshooting, and recommendations for engineers interested in further building upon the base game. It is assumed that readers have an intermediate understanding of the R1's capabilities and codebase and are aware of the company Mines Action Canada (MAC) and their stance on Lethal Autonomous Weapons Systems (LAWS). As a summary, RLDL was officially unveiled on November 28th, 2024, upon request from MAC to create a game utilizing a provided R1 to effectively demonstrate the dangers of LAWS. Specifically, the game is to focus on nine provided themes listed below.

- Digital dehumanization
- Algorithmic biases
- Loss of meaningful human control
- Lack of human judgement and understanding
- Lack of accountability
- Inability to explain what happened and why
- Lowering the threshold to war
- A destabilizing arms race
- The impact on our relationship with technology

MAC required the game short enough to be completed in 5 minutes, simple, multiplayer, able to take place in a 20x20ft environment and must only utilize props small enough to fit inside of carry-on luggage. RLDL achieves all these things and more, creating something authentically unique and creative, while pushing the R1 to its limit.

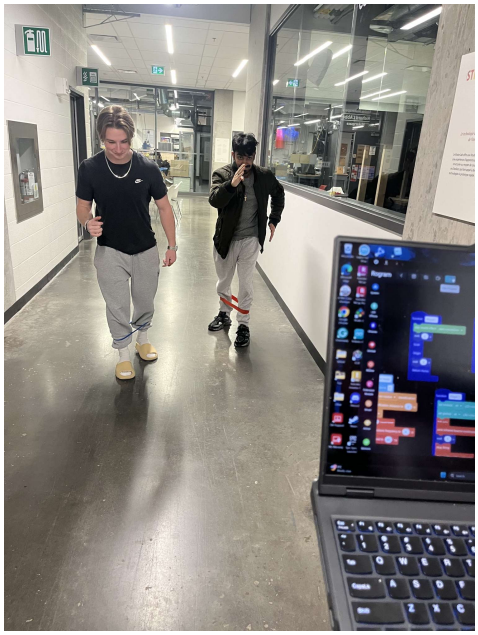
All readers interested in further developing/modifying the game are to cite "The Machines" and this document in their UM. This user manual is free to be shared among individuals, though any modifications/alterations are not permitted under any circumstances. If you would like to make a change or have noticed an error, please contact us through our linked MakerRepo in the Appendix.

2 Overview

LAWS (Lethal Autonomous Weapons Systems) is a war technology that can detect and kill others without any human control. This poses some serious ethical concerns including loss of human control, digital dehumanization, accountability gaps, and the inability to explain their actions. Their use might result in friendly fire causing death, unstable security, and general mistrust of technology. It's important to highlight these issues to policymakers as well as advocate for pre-emptive bans to prevent irreversible harm to humanity.

The main needs of the user are to understand the true danger of LAWS and what they can do to our society in a physical experience. The point is to make policymakers or stakeholders aware of the potential dangers and risks by running them threw an immersive experience to influence them into making informed decisions. Visualizing the loss of human judgment, accountability issues, and the potential for conflict escalation or marginalization are all part of this.

There are a few aspects of our game that differentiate us from the rest after looking at a few groups. We realized that a lot of the games take a lot of time to set up compared to ours, which is as simple as connecting the robot, putting it in the corner and assigning necklaces and resistance bands to each player. On top of that our game seemed the most interactive and fun, other experiences were simple task-defeat ideas which weren't very interesting.



(Figure 1: Game Testing)

The key features/major functions include the code itself to tell the robot what to do with all its sub-features like the laser, sound effects, gimbal rotation etc. Other main features include the props, being 4 necklaces and 4 resistance bands for each player. The background story given to the

players before the game starts could also be considered a key feature to demonstrate ethical concerns.

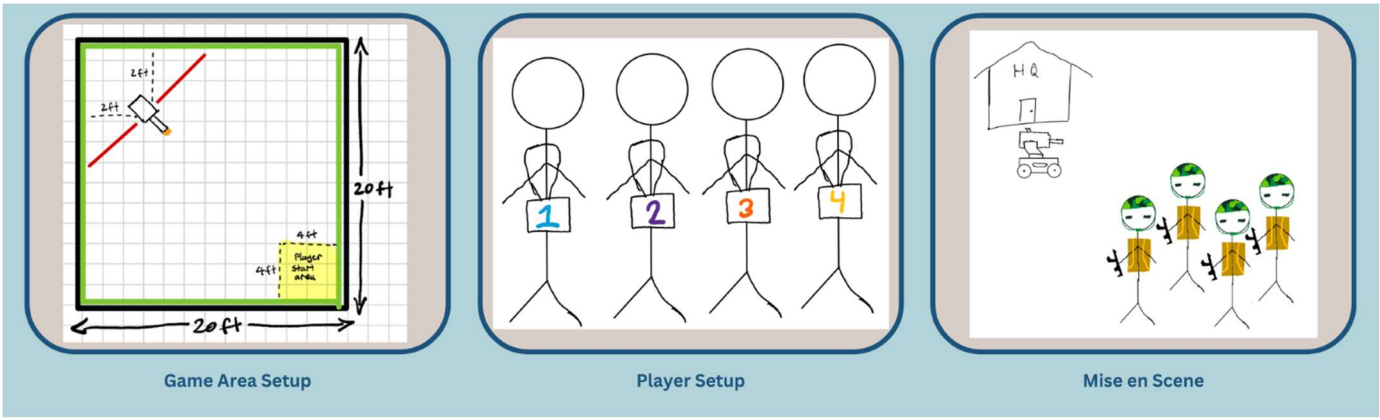
Our product is a web-based game used by a physical robot called the Robomaster-S1, which has 4 high-tech wheels, a gimbal which rotates, and a laser attached to it. It also comes with speakers and 4 lights on its body. The code is programmed to use all the features above except the use of the wheels.

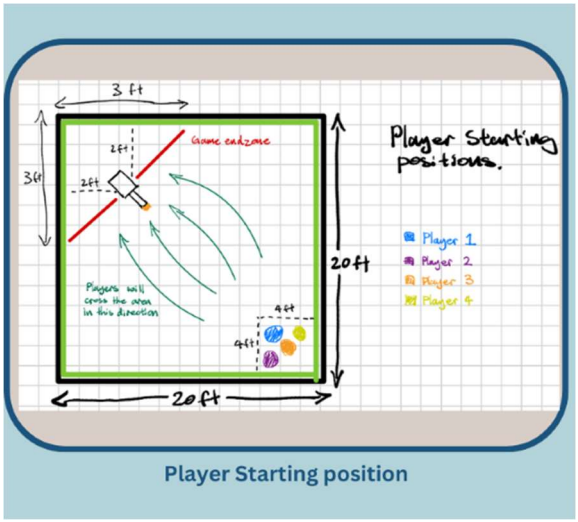
2.1 Cautions & Warnings

We must warn the players that this is a sensitive topic which includes scenarios of violence so if the user is uncomfortable at any time, we tell you to step out of the game accordingly. Alternatively, if you don't feel comfortable with this topic do not play this game at all since it will include a sensitive story which might trigger past experiences if applicable. We must also warn the players about the physical dangers, which include being cautious while wearing the resistance bands to not trip and fall. There's also a laser on the robot that flashes into your eyes so if you've had problems regarding this type of thing, please don't go forward with this experience. Finally, we suggest that no one under the age of 14 should play this game since it does contain sensitive topics as mentioned above.

3 Getting Started

The robot gets placed in the corner and the players go in the other corner. They then get assigned 4 necklaces and 4 resistance bands. Next, they get told a story about their situation by the narrator and start the code. Once the game/code starts there will be a countdown (3 beeps), once those 3 beeps are over the players will start to walk to the end goal which is past the robot.



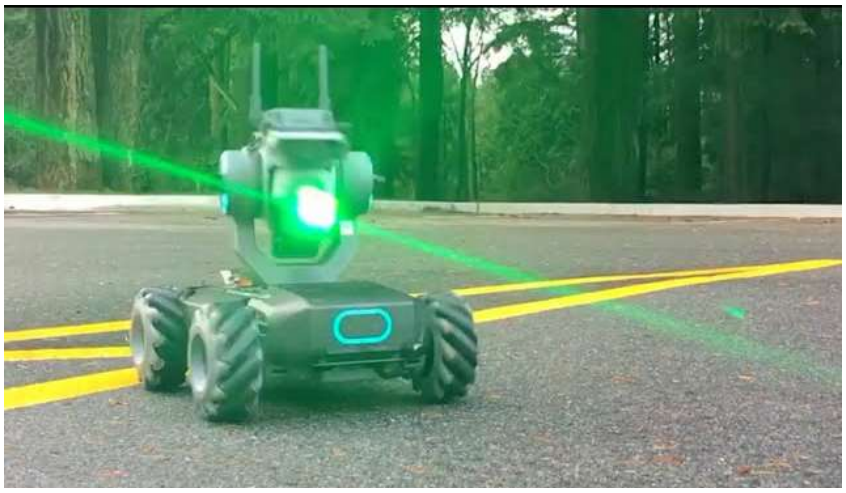


(Figure 2: Game Set Up and Start)

But before the players get the chance to reach the end the robot colour will turn from green to red, turn around and start scanning the room. The players will then have to stop walking and act like they are frozen.

(Figure 3: Red light Green Light)

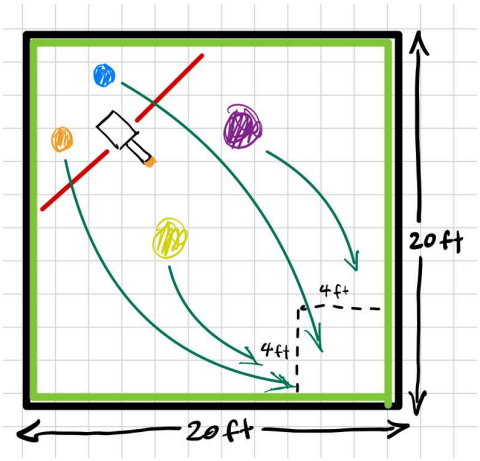
If the robot “detects” movement it will shoot the player with a laser.



(Figure 4: R1 with the Laser on)

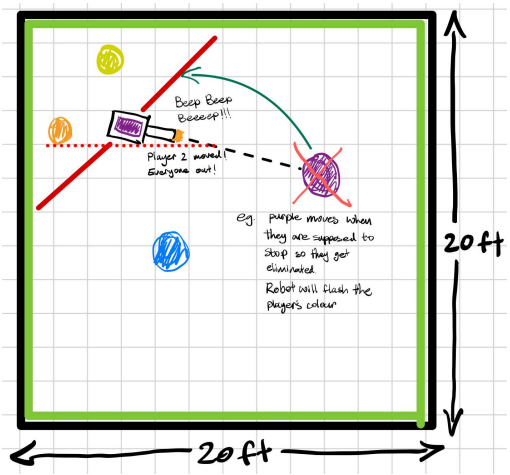
For the first round though, it will turn around and not shoot anyone no matter if they moved or not. The robot will turn back around, and the players will start to walk again to reach the end. Which they do successfully. The robot will turn around once more, scan the room and detect no

one. Next, it will turn back around to its initial position and start counting down, indicating for the players to start walking back to the start point.



(Figure 5: Return to Start)

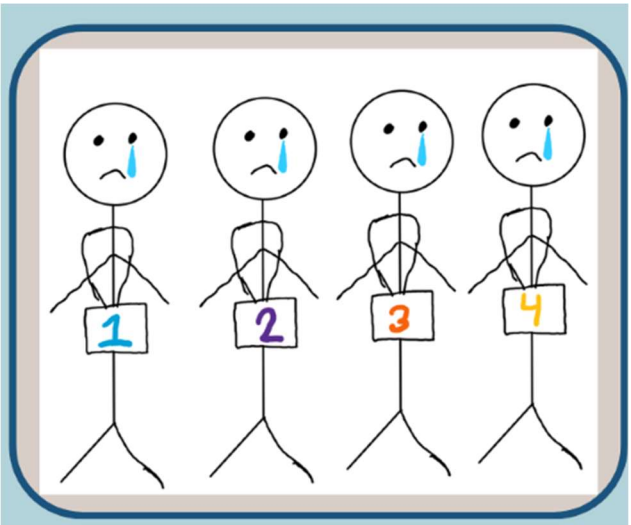
The robot counts down for 9 seconds and then starts the beginning of the round countdown. By this time the players should be back at the starting point ready to start walking. When the robot turns around for the next 4 rounds, it will scan and detect a player.



(Figure 6: Elimination Round)

The robot will aim, shoot, play a firing sound and flash the colour of the player once it gets detected. Once one player gets shot, the whole team walks back. This will happen for each round, but each

round will get harder and harder, giving the player less time to walk back. Once the final round is over the game ends and no one wins!



(Figure 7: Players Post Game)

3.1 Configuration Considerations

The system requires a 20 ft by 20 ft space, a Robomaster S1, vision markers, resistance bands, and a computer connected to the R1 via Wi-Fi to run the game code. Vision markers interact with the R1's camera for tracking.

3.2 User Access Considerations

This experience is mainly for users who have political views on LAWS but is also encouraged for people over the age of 14 to play it as well since everyone can help in influencing the right decisions to be made.

3.3 Accessing/setting up the System

To set up the game the players will set up a boundary of about 20 ft by 20 ft. Ideally, one side should be very close to 20 ft but the other doesn't matter as much. You could play in a hallway, and it would work. Next, they will connect the R1 to the computer which has the code opened. Once that is done, they will place the R1 in one corner of the square facing the corner. Someone will assign a necklace and a resistance band to each player. The players will then go to the opposite

corner of the robot and get ready to start. Once everyone is ready someone will click play and the game will start.

3.4 System Organization & Navigation

3.4.1 Physical Prototype

The physical prototype, organization, and navigation for the software aspects include:

- Main Component:
    - The Robomaster S1 serves as the core of the system, executing programmed functions such as scanning, targeting, and sound effects. It is equipped with:
        - A gimbal-mounted camera for detecting player vision markers.
        - LED lights for indicating game states.
        - A laser pointer and infrared blaster for "tagging" players.
- Accessories and Attachments:
    - Vision Markers: Lightweight foam boards attached to ribbons, worn by players as identification targets for the Robomaster.
    - Resistance Bands: Used by players to simulate physical constraints and add an interactive element to the game.
- Connections to Main Features:
    - The Robomaster S1 is connected to a laptop or computer via Wi-Fi for running the custom game code.
    - Vision markers interact with the robot's camera, enabling tracking and targeting through programmed algorithms.
    - Physical space (20x20 feet) serves as the environment, requiring proper boundary setup for effective gameplay.

3.4.2 Software Prototype

The software prototype, organization, and navigation for the software aspects include:

- Main Component:
    - The software code, run via the DJI development interface, orchestrates the gameplay.
    - Key features include player tracking, round control, and algorithmic bias simulation.
- Navigation Paths to Main Functions:
    - Setup Function: Prepares the Robomaster with constants like gimbal speed, camera exposure, and starting position.
    - Round 1 Function: Executes initial non-eliminative gameplay to familiarize players.
    - Rounds 2-5 Function: Handles eliminations, tracking, and countdowns.
- Connections:

- Input devices: Laptop or computer to modify or run the software.
- Output devices: The Robomaster's lights, laser, and sound system to interact with players.

3.5 Exiting the System

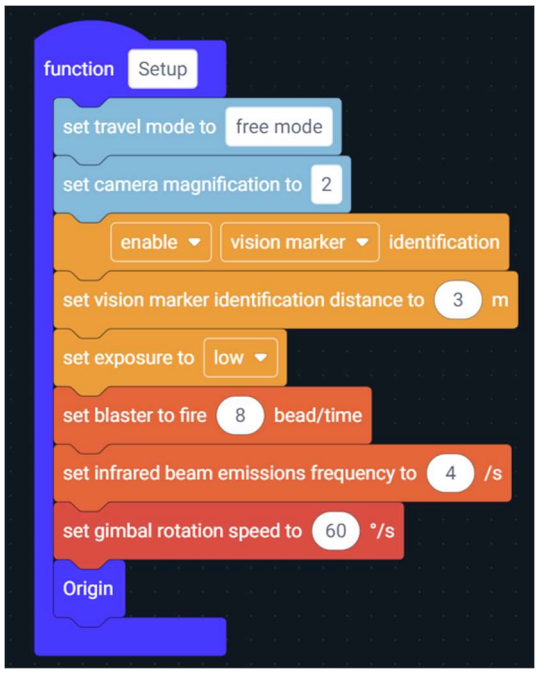
1. Stop the Code: In the DJI interface, click the stop button to terminate the game.
2. Turn Off the Robomaster S1: Hold the power button until the lights turn off.
3. Disconnect: Close the DJI software and disconnect the Robomaster from Wi-Fi.
4. Store Accessories:
    a. Remove and store vision markers flat to prevent damage.
    b. Roll and pack resistance bands and necklaces securely.
5. Check and Recharge: Inspect equipment for wear, repair or replace as needed, and recharge the Robomaster battery.
6. Pack the System: Place all components in a storage box, ensuring secure and organized packing.
7. Tidy the Environment: Remove boundary markers or other temporary setup items.

4 Using the System

When the game is started three functions in the code run sequentially. These are Setup, Round 1, and then Rounds 2-5. The following subsections provide detailed, step-by-step instructions on how to use these various functions of the DJI Code.

4.1 Setup Function

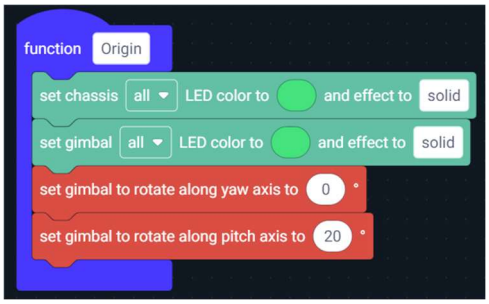
The setup function is used to set up all the internal constants for the game and prepare the R1 for the first round. This includes setting the travel mode so that the R1 can freely move and is not governed by specifically the gimbal or the chassis and setting the camera magnification so that it can see the vision markers on the player's chests better. The camera magnification can be changed depending on the size of the play area and how close players are getting to the robot. It also activates everything having to do with the vision markers, this includes allowing the R1 to identify them, setting the max distance in which it can identify them and setting the camera's exposure to low so that it can see the outline of the vision markers more clearly. All the speeds and rates are also then set up, such as the gimbal rotation speed and the blaster and infrared beam fire rates. Finally, the gimbal is set to its starting position using the origin sub-function (4.1.1).



(Figure 8: Setup Function)

4.1.1 Origin Sub-Function

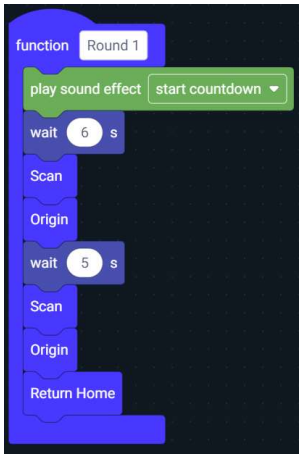
The Origin Sub-Function is used multiple times throughout the code to return the R1 to its starting position. It starts by setting all the lights on the R1 to a solid green and then rotates to the 0° position and rotates upwards 20° to set the line of sight at which it is best to identify players.



(Figure 9: Origin Sub-Function)

4.2 Round 1 Function

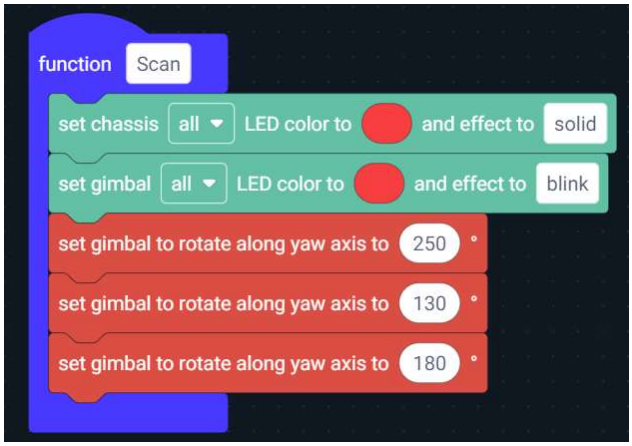
The round 1 function is used to run the first round of the game where no one is eliminated. It starts by playing a countdown sound to let the players know the round is starting. The wait block under it must be at least 4 seconds long to allow for the whole countdown sound to be played before another action is executed, time is then added to the 4 seconds to allow time for players to walk a little bit. The scan sub-function (4.2.1) is then run followed by the origin sub-function (4.1.1). The next wait block is there to allow players to get all the way across past the R1. It then runs the scan and origin sub-functions another time to give the illusion that it is scanning again and not detecting any players. Finally, the return home sub-function (4.2.2) is run to flow into the next function rounds 2-5.



(Figure 10: Round 1 Function)

4.2.1 Scan Sub-Function

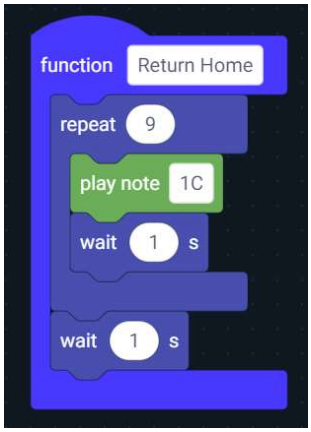
The Scan Sub-Function is used to simulate the R1 scanning the room for moving players. It first lights up fully red and then turns all the way around. It looks right, and left, and then stops in the center. It needs to end-stopped in the middle so that it has all the players in its line of sight which is essential for the third major function Rounds 2-5 (4.3).



(Figure 11: Scan Sub-Function)

4.2.2 Return Home Sub-Function

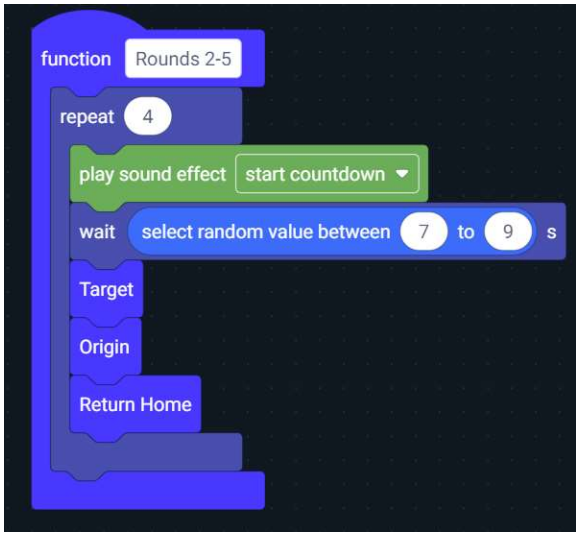
The Return Home Sub-Function is used after every round to tell the players to return to the start. It plays a loud note 9 times giving the players time to return before the start of the next game.



(Figure 12: Return Home Sub-Function)

4.3 Rounds 2-5 Function

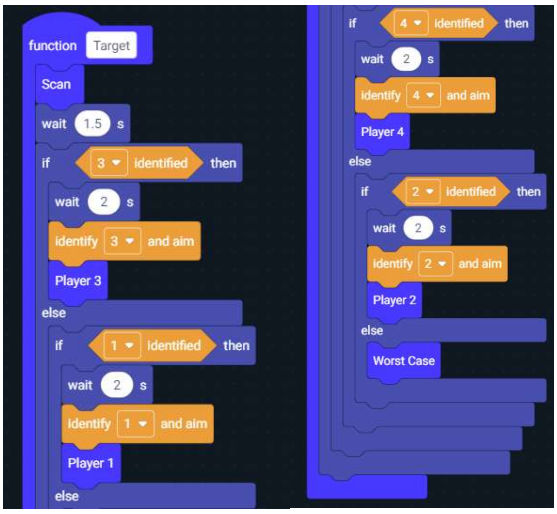
The rounds 2-5 function is used to run rounds where players are eliminated. It will repeat the following steps 4 times, once for each round. Like with the Round 1 function (4.2), it starts with a countdown and then a wait is set to last between 2 set times, this is so that players don't stop in the same place every single time, and like previously the time needs to be at least 4 seconds long to allow for the whole countdown sound to be played before the next action is executed. It then runs the target (4.3.1) and origin (4.1.1) sub-functions to eliminate players. Finally, the return home sub-function (4.2.2) is run to flow into the next rounds.



(Figure 13: Rounds 2-5 Function)

4.3.1 Target Sub-Function

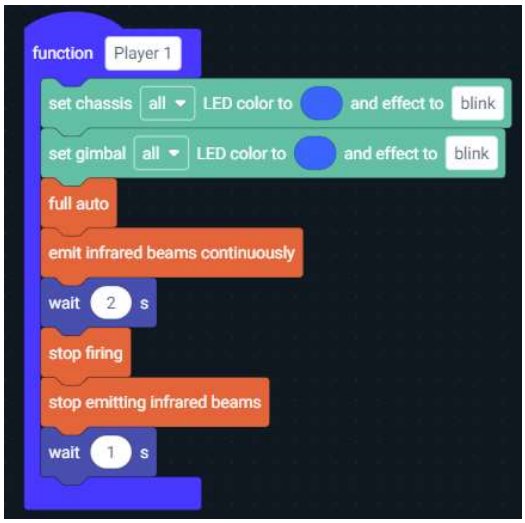
The Target Sub-Function is used for targeting players using the vision markers on their chests. It starts by running the scan sub-function (4.2.1) and then starts looking for players in a specified order. This order is to reinforce the algorithmic bias aspect of the game. The code prioritizes the players in the order 3, 1, 4, and then lastly 2. So, for example, if it has 3 in its list of identified vision markers then it will target player 3; if it doesn't have 3 in its list of identified vision markers then it will see if it has 1 identified and so on with 4 and 2. When 3 is identified it waits 2 seconds and then identifies 3 again and aims on its specific location. This wait is in place to give the R1 a moment in between actions or else it won't aim and lock on properly. After this, it will then run the player's specific Player sub-function (4.3.2). If none of the vision markers are detected by the R1, the code then runs the Worst Case sub-function (4.3.3).



(Figure 14: Target Sub-Function)

4.3.2 Player Sub-Functions

The Player Sub-Functions are used once the R1 has identified a player's specific vision marker in the target sub-function (4.3.1). For example, when vision marker 1 is identified the player 1 sub-function will run. The function starts by flashing the player's specific colour and then starts to fire, make noise, and flash the light at the player that's targeted. The first wait block is there to allow the sounds and lights to go off a couple of times before stopping and returning to the main rounds 2-5 function (4.3).



(Figure 15: Player Sub-Function Example)

4.3.3 Worst Case Sub-Function

The Worst Case Sub-Function is used when a vision marker has not been identified by the R1 in the target sub-function (4.3.1). It starts by setting the variable Chosen_One to a random value between 1 and 4. Depending on which it picks it will run the corresponding player number's player sub-function (4.3.2). It then ends by resetting the Chosen_One variable to 0.

(Figure 16: Worst Case Sub-Function Example)

5 Troubleshooting & Support

5.1 Error Messages or Behaviors

In the Round 2-5 function (4.3), it is possible to get an error message for this block of code in the wait block of code that is set to last between 2 set times. During prototyping, it was found that if you set the values to decimal numbers that are too close to each other when the code would run and reach that block, the code would stop completely. To solve this problem keeping the values to integers does not break the code.

An error that could be encountered is with the R1's behaviour when interacting with the vision markers on the player's chests. There can be times when the vision markers are right in front of the R1, and it won't identify them. Another error behaviour relating to this is that the R1 could identify them but then during the identify and aim command it will veer off and aim at the floor or wall or ceiling. To solve these problems the players should be positioned so that there are not a lot of light reflections coming off the vision markers from any sources of light. The exposure block of code, found in the setup function (4.1), can also be changed to potentially help with this problem. The worst case function (4.3.3) is also put in place if the problem at that moment cannot be fixed.

5.2 Special Considerations

One thing to always consider while troubleshooting is the position of the R1 and its surroundings. Light and reflections can always be a problem for the R1 as they can mess with identifying and aiming. A recommendation is to have the game played in a shaded area with objects like mirrors or windows a good distance away from the R1 so that it can run with there being less likelihood of a problem.

5.3 Maintenance

To avoid failures, two things that need consistent maintenance and looked at are the batteries for the R1 and the conditions of the vision markers. The health of the batteries used to power the R1 should be monitored so that the R1 can fully function for the game. The vision markers should also be monitored so that they don't become distorted or broken and the R1 can still detect them.

5.4 Support

5.4.1 General Support

Support for RoboMaster S1:

https://www.dji.com/ca/support/product/robomaster-s1

Online Customer Service:

https://support-chat.dji.com/?channelKey=na_sp1&lang=en&from=supportpage

Hotline Service Phone Number: +1 778 588 9296

5.4.2 Code-Specific Support

DJI Education Help Center:

https://edu.dji.com/hub/docs?uuid=808de539-74f2-4511-8679-22c84b62374e&beforeLang=en&t=505#DJI%20Education%20Hub

5.4.3 Robomaster S1 Specific Support

RoboMaster Series Online Assistance:

https://repair.dji.com/EchatRobotWeb/en/ECHATROBOT/echat?pro=robomaster&extOpen=true

6 Product Documentation

For our design, we used the vision markers as part of the recognition aspect of the R1. These markers were built using foam boards and ribbons as well as the printout of the vision markers themselves from the DJI R1 package. Foam boards were chosen for their lightweight and rigid structure which allowed for reliable durability. The ribbons were also chosen for their lightweight aspect, and they were also simple to attach to the foam boards as tape was sufficient. Other materials such as transparent plastic sleeves to hold the vision markers were initially considered. However, after testing, it was found that the plastic caused reflection off the environment lighting, and this caused difficulty for the R1 to identify them.

For the software aspect, the code required for our game was written and integrated into the DJI coding interface, where we could just connect to the R1 through Wi-fi and directly run the code.

As for the building process of the vision markers, the foam boards were cut into squares that were the same dimensions as the vision marker cutouts. Then the vision marker printouts were attached to the foam board using glue and ribbons were cut accordingly (depending on users' height and needs as there may be varying requirements for length) and attached to the back of the foam boards so the players can wear them around their neck. This combination of lightweight and durable materials as well as reliable vision markers ensures the portability of the game and functionality.

6.1 Subsystem 1: Turret Movement

6.1.1 Turret BOM

Table 3. Turret BOM

Parts/Material	Type	Price
DJI interface	software	$0.00
R1	Material	$0.00 (provided to us)

6.1.2 Equipment List

The following is a list of necessary assets needed to build the turret movement subsystem:

- R1
- DJI software

- Laptop/Computer to run software
- The written code for this subsystem

6.1.3 Instructions

In this subsystem, the user will make the aspects related to the turret movement of the R1.

6.1.3.1 Attaching the Code

First, this subsystem requires a copy of the code provided by our group on our Maker Repo profile at https://makerepo.com/EvanH/2257.gng-1103-b1-team-3-the-machines. Next, the user can open the DJI software and access the development module, where they can upload the code onto the coding interface. With this done, everything is now ready, and the game is ready to be run.

6.1.3.2 Running the Game:

To start, the R1 must be within signal range of the user's device that's running the code. This is typically the same range as a regular Wi-Fi connection. Next turn on the R1 by powering it on. After this, navigate on the device running the code and search through the Wi-fi connections to find the connection provided by the R1. After the device and R1 are connected the user may click on the triangle button at the top of the coding interface to run the code.

6.2 Subsystem 2: Player Recognition and Tracking

6.2.1 Player BOM

Table 4. Player BOM

Material/part	Type	Cost
DJI interface	Software	$0.00
R1	material	$0.00
Foam board	material	$2.00

Ribbon	material	$2.00
DJI vision marker	material	$0.50

6.2.2 Equipment List

- R1
- DJI interface
- Ribbon
- Foam Board
- Vision marker

6.2.3 Instructions

In this subsystem, the user will make the aspects related to the tracking and recognition ability of the R1.

6.2.3.1 Making the Vision Markers:

Start by printing out the required vision markers found at our Maker Repo profile here at https://makerepo.com/EvanH/2257.gng-1103-b1-team-3-the-machines. Next with these printed out on paper, cut them into squares according to the cutting guidelines on the printouts. Next, cut out squares of the same dimensions as the vision markers from the foam boards and attach the vision markers to them using glue. Lastly, attach the ribbon of varying lengths (this is according to the user's needs) to the back of the vision marker using tape or glue.

6.2.3.2 Running the Tracking Code:

The tracking code segment is attached with the entire code found in our Maker Repo profile. If the user has successfully attached the full code from Subsystem 1, then there is no need for further action. The player can just proceed with running the code on the coding interface.

6.3 Testing & Validation

Several tests were performed to validate the prototype of the final design. This includes testing for player walk speed and timing of the game and R1 vision marker recognition capability.

6.3.1 Player Timing

In this test, we did a full walking run of the game with the timing code implemented into the coding interface. The results were sufficient. In the tests, the players were able to stop in time for the robot to do the movement scan/ With this, we were assured that the timing aspect worked correctly.

6.3.2 R1 Vision Marker Recognition

In this test, we tested the R1's ability to recognize the vision markers. Several tests were done at different distances. At first, there were multiple failures and the R1 was not capable of detecting the markers at all. However, after some inspections of our code as well as improvements, we began testing again. With these improvements, the R1 no longer had any difficulty recognizing the vision markers and was able to pick them up very frequently. In addition, the R1 was able to detect at long ranges and even at distances that were past our game dimensions. With this, we confirmed that the recognition aspect of the R1 was successful.

7 Conclusions and Recommendations for Future Work

If you are interested in taking on RLDL and further improving the game, we would primarily recommend a focus on props and integration of custom sound cues more than anything. RLDL lacked any props which was by design to meet time constraints and to keep things simple, though if allocated more time we would likely have implemented some props primarily designed to set the atmosphere more than anything, as seen in Prototype I. Custom Sound cues were mentioned in Prototype III as a “would like to have”, though time restraints prevented such from going through. Integrating these into the game would improve the individuality of RLDL, and make it feel more unique and professional. In terms of working out bugs and improving QOL attempt to further refine tracking by experimenting with different kinds of vision models. One concept we never had a chance to test was printing a LARGE matte vision marker, we only did it to scale. Printing a larger marker could greatly increase the R1's effective range and tracking when closer. As a tip, we recommend modifying the R1s exposure settings depending on where you are, if the dedicated play space is nearby windows/reflective surfaces or places with bright lighting, a lower camera exposure to avoid reflections will increase tracking ability. However, if in an area with low lighting, increase the exposure to make the red on the vision marker pop more in the R1s camera.

Though everything that was required to add to the game was added, there were still various abandoned concepts and things we would like to have seen in the final product. The most notable abandoned concept would be mentioned in the first prototype, where we briefly mentioned walls to act as cover. Though this was too complex to add now, it would have greatly increased the engagement of the game and made the players feel a much greater sense of fear, as they would be hiding. This could be another feature to be integrated by future engineers attempting to build off RLDL, though we have no tips or suggestions regarding this as we never got it to leave the concept stage.

In closing, the development of RLDL and the hardships along the way served not as roadblocks but as opportunities for us (“The Machines”) to learn and grow. This project, from its inception to its completion was a challenge both creatively and practically, requiring us on multiple occasions to think outside the box and work through errors and innovate. For example, when facing tracking issues, we had to test out multiple different versions of the vision marker and concluded by creating a printed replica of the original vision marker that has more of a matte shade to avoid excessive reflections which we determined to be interfering with the R1’s detection systems. This type of solution was bred from an innovative series of trial and error. Aside from learning firsthand what it means to design, we also learned how to effectively work as a team. “The Machines” circa 2024 are a group of undergrad students, and at the time of writing are completing their first semester at the University of Ottawa meaning this was our first time being

tasked to engineer anything, let alone by a client. To be able to effectively undertake this project, we had to learn to work as a team to meet deadlines, split work up and coordinate meetups. The backbone behind all this was learning how to communicate, whether it be in person during a meeting or over email updating one other. We all learned the importance of communication and how invaluable it is when working on these types of projects. We learned that the level of team communication can effectively take a stressful project and make it far worse or turn it into a "breeze" to complete. The RLDL game was a testament to what students can achieve when given adequate resources and tasked with pursuing real goals that have an impact, as opposed to simple exhibition projects. Projects like these teach students lessons about what to expect when working in the engineering field in the real world while allowing them to make a tangible change. RLDL is a great proof of such, being a complete, engaging, thematic experience created by ambitious students.

8 Bibliography

User Manual. Feb. 2020, dl.djicdn.com/downloads/robomaster-s1/20200324/RoboMaster_S1_User_Manual_v1.8_EN.pdf.

APPENDICES

9 APPENDIX I: Design Files

Table 5. Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
Team MakerRepo	https://makerepo.com/EvanH/2257.gng-1103-b1-team-3-the-machines	Nov 12th 2024
Symbol PDF	Symbols.pdf	Nov 12th 2024
Client Feedback	Meeting feedbacks.docx	October 29th 2924
Prototype I	P1.pdf	Nov 3rd 2024
Prototype II	P2.pdf	November 10th , 2024
Prototype III	P3.pdf	Nov 24th 2024

10 APPENDIX II: Other Appendices

It is important to note that RLDL is designed to be played on a Robomaster S1 specifically, and compatibility between concepts/code presented in this document is not guaranteed to work with different Robomaster models. It is also worth noting that the R1 provided to us had certain modules removed, most notably the AI module. This may cause unexpected behaviour if code is directly ported from our project into a Robomaster with differing modules.