Revision Date:	September 15, 2024
Revision:	A
Document Num:	RLT-001
Project:	Robot Laser Tag
Course:	CMPSC-488-002
Team:	2

Discovery Document

Penn State Abington

1600 Woodland Rd, Abington, Pa 19001

Joseph Porrino, Aaron Feinberg, Mansib Ahmed, Katherine Banis, Clay Blockinger, Fernanda Martins

Table of Contents

1 Document Responsibility & Revision Tables

1.1 Revision table

Rev	Date	Section	Author	Description of Changes
1.00	09/02/24		Aaron Feinberg Mansib Ahmed Joseph Porrino Clay Blockinger Katherine Banis Fernanda Martins	Initial Document Creation

1.2 Document Responsibility Table

Name	Section(s)
Aaron Feinberg	Document Template, User Stories, High Level Overview
Mansib Ahmed (Syed)	High Level Overview, Project Objectives
Katherine Banis	Team Members, Glossary, Mission Statement, Works Cited
Clay Blockinger	Technical Requirements
Joseph Porrino	Document Template, User Stories, Functional Requirements, technical requirements
Fernanda Martins	Assistant

2 Objectives & Mission Statement

2.1 Project Objectives

The objective of this project is to design and build a fully autonomous robot capable of participating in a laser tag game with two other robots. The project will integrate hardware, software, and 3D printing technologies to ensure that the robot can adhere to the game rules, navigate the playing field, and interact with other robots effectively. Successful completion will depend on effective collaboration among team members, timely completion of milestones, and rigorous testing to ensure optimal performance.

2.2 Mission Statement

Our mission is to design and build a fully autonomous robot that demonstrates teamwork and competitive strategy in a simulated laser tag environment. By integrating advanced technologies in machine vision, navigation, and artificial intelligence, we aim to develop a system that can interact dynamically with both physical and virtual obstacles. Through collaboration, innovative engineering, and thorough testing, we will create a robot that can adapt to different environments, outmaneuver opponents, and achieve victory in a challenging and immersive game scenario.

3 Stakeholders & Team Members

3.1 Stakeholders

Client: Joe Oakes

Project Manager: Zachary Whitaker

3.2 Team Members

3.2.1 Contact Information and Aliases

Aaron Feinberg – ajf5972@psu.edu

Joseph Porrino – jhp5207@psu.edu

Katherine Banis – kbb5551@psu.edu

Mansib Ahmed (Syed) – sma6629@psu.edu

Clay Blockinger – crb6023@psu.edu

Fernanda Martins – fgm5045@psu.edu

3.2.2 Team Member Positions & Responsibilities

Aaron Feinberg

As team leader, he ensures that all project phases proceed smoothly, and milestones are achieved on schedule. Aaron manages all team communications, coordinating through Discord, Microsoft Teams, and Canvas for meetings, discussions, and formal submissions. He will be involved in developing and optimizing complex algorithms for environment mapping, object recognition, path planning, and target identification. He will also participate in the assembly of the robot, contribute to the coding for navigation and sensor integration, and assist in the setup and calibration of various sensors.

Joseph Porrino

Joseph is responsible for handling all project documentation, including progress reports and preparation for final submission materials. He will be focused on image processing and neural network creation for predicting the robots movements.

Katherine Banis

Katherine will play a critical role in developing algorithms that enhance the robot's ability to map environments, recognize objects, plan paths, and identify targets. Her technical contributions extend to participating in the robot's assembly and coding for its navigation and sensor integration. Katherine will also support the team in setting up and calibrating the necessary sensors to optimize the robot's functionality.

Mansib Ahmed

Syed will focus on algorithm development for mapping, recognition, and navigation tasks that are central to the robot's functionality. He will take part in the assembly of the robot, coding for its navigation systems, and integrating various sensors. His role also includes assisting in the calibration and optimization of sensor data processing, vital for the robot's performance.

Clay Blockinger

Clay is tasked with designing and 3D printing the structural components of the robot, ensuring durability and functionality. In addition to his design duties, Clay will develop algorithms for various operational aspects of the robot and lead in its assembly. He also will contribute to coding for navigation, help with sensor integration, and assist with the calibration process, ensuring all components work together seamlessly.

Fernanda Martins

Fernanda is integral in developing and fine-tuning algorithms for environment mapping, object recognition, path planning, and target identification. She will participate in the robot's assembly and contribute to the coding efforts for its navigation and sensor integration. Fernanda's role also includes assisting with the setup and calibration of sensors, ensuring the robot operates efficiently and effectively in its designed tasks.

4 Robot Laser Tag – Project Description

4.1 High Level Overview

The purpose of this project is to build and program autonomous robots that simulate a laser tag game. Two robots with a corresponding target will be positioned on either side of the course (also referred to as the “battlefield”). The game’s goal is to score the maximum number of points by hitting the opposition robot’s target while avoiding its laser attacks and other obstacles placed throughout the battlefield. The game will also feature a third robot known as the disrupter that will obstruct both robots on their pathway to victory. In general, all three robots can switch to the disrupter role. Robots will operate under fully autonomous control in each role.

The battlefield will be an enclosed space with obstacles which are immovable. The course will feature a Lidar sensor in the center providing a continuous circular scan of the course and providing this information to all team’s robots by means of a RESTful web API.

5 User Stories

This section captures the requirements of the end user of the robot.

5.1.1 Object Detection

5.1.1.1.1 Object detection **shall** distinguish between other robots and obstacles

5.1.1.1.2 Object detection **shall** detect walls/boundaries

5.1.1.1.3 Object detection **shall** detect enemy target

5.1.2 Navigation/Planning

5.1.2.1.1 Navigation/Planning **shall** maneuver robot into a shooting position to hit the target

5.1.2.1.2 Navigation/Planning **shall** navigate robot around obstacles

5.1.2.1.3 Navigation/Planning **shall** avoid other robots

5.1.2.1.4 Navigation/Planning **shall** keep robot within bounds

5.1.2.1.5 Navigation/Planning **shall** use objects to block enemy robot laser

5.1.3 Shield

5.1.3.1.1 Shield **shall** be able to block enemy laser from hitting target

5.1.3.1.2 Shield **shall** have a cooldown time after being used

5.1.4 Laser

5.1.4.1.1 Laser **shall** be able to aim and fire at other robots and targets

5.1.4.1.2 Laser **shall** have a cool down period after firing

5.1.5 Hit Detection

5.1.5.1.1 Hit detection **shall** detect laser on its sensor surface

5.1.5.1.2 Hit detection **shall** temporarily disable robot when hit with laser

5.1.6 Database logging

Robot will have access to this database to update its strategy

5.1.6.1.1 Database logging **shall** log detected hits

5.1.6.1.2 Database logging **shall** log firing of laser

5.1.6.1.3 Database logging **shall** log detection of target

5.1.6.1.4 Database logging **shall** log firing at target

5.1.6.1.5 Database logging **shall** log shield engagement

5.1.6.1.6 Database logging **shall** log shield disengagement

5.1.6.1.7 Database logging **shall** accept and respond to queries

5.1.7 Disruptor Role

5.1.7.1.1 Disruptor role **shall** find the other robots and hit their target with the laser

5.1.7.1.2 Disruptor role **shall** physically present itself as an obstacle to other robots

5.1.7.1.3 Disruptor role **shall** change strategy based on the state of the game (ex: If a team is up in points, focus on disrupting that team)

6 Functional Requirements

This section defines the behavior, features, and functions that the robot must have to meet the needs of the users and stakeholders.

6.1 Object Detection

6.1.1.1.1 Object detection **shall** use M5 Core S3 camera for site

6.1.1.1.2 Object detection **shall** use infrared emitter to measure distance

6.1.1.1.3 Object detection **shall** use machine vision to identify objects.

6.1.1.1.4 Object detection **shall** use machine vision to distinguish robots, targets, and obstacles.

6.1.1.1.5 Object detection **shall** use Jetson Nano to run machine vision program

6.1.1.1.6 Object detection **shall** use a lidar in the center of the map for preemptive object detection.

6.2 Navigation/Planning

6.2.1.1.1 Navigation/Planning **shall** use two Vex Motor modules to move the wheels.

6.2.1.1.2 Navigation/Planning **shall** use PWM to direct the Vex Motor modules

6.2.1.1.3 Navigation/Planning **shall** use M5 Core S3 IMU for present direction.

6.2.1.1.4 Navigation/Planning **shall** use a neural network to predict the next movement of the robot.

6.2.1.1.5 Navigation/Planning **shall** translate neural network output to PWM

6.2.1.1.6 Navigation/Planning **shall** use machine vision, IMU, infrared, and lidar output for neural network input.

6.2.1.1.7 Navigation/Planning **shall** use Jetson Nano for computational processing.

6.3 Shield

6.3.1.1.1 Shield **shall** use a servo to move in front of and away from laser sensor.

6.3.1.1.2 Shield **shall** use PWM to move servo.

6.3.1.1.3 Shield **shall** track when PWM signal is sent to servo to start cooldown period.

6.3.1.1.4 Shield **shall** use machine vision and lidar output for neural network input.

6.3.1.1.5 Shield **shall** use neural network output to predict when to shield laser sensor.

6.4 Laser

6.4.1.1.1 Laser **shall** use two servos to move in the x and y direction.

6.4.1.1.2 Laser **shall** use PWM to adjust servo position.

6.4.1.1.3 Laser **shall** use machine vision to identify enemy target.

6.4.1.1.4 Laser **shall** translate image coordinates to real world coordinates.

6.4.1.1.5 Laser **shall** use Jetson Nano for computational processing.

6.5 Hit Detection

6.5.1.1.1 Hit detection **shall** increase hit count when sensor detects a laser.

6.5.1.1.2 Hit detection **shall** stop Vex Motor from receiving commands from Navigation/Planning.

6.6 Database logging

6.6.1.1.1 Database logging **shall** use SQL database to log data

6.7 Disruptor Role

6.7.1.1.1 Disruptor role **shall** be activated via button on Raspberry Pi screen.

6.7.1.1.2 Disruptor role **shall** use Navigation/Planning data to disrupt other robots

6.8 Power

6.8.1.1.1 Power **shall** use a USB power bank that connects to Jetson Nano and M5 Core S3

7 Technical Requirements

7.1 Hardware System Requirements

7.1.1.1 Jetson Nano

Function: Image processing, object detection, and obstacle avoidance.
How:
- The Jetson Nano will run PyTorch generated deep learning models for predicting the robot's next move.
- The Jetson Nano will receive image and IMU data from the M5 Core S3 via a USART connection.
- The Jetson Nano will process image data using OpenCV
- The Jetson Nano will run separate python processes for image processing and I/O operations.
  - A pipe will be used to send data between processes.
- The Jetson Nano will use a GPIO header to connect Vex Motors, laser servos, and shield servos.

7.1.1.2 M5 Core S3 (Running MicroPython)

Function: Peripheral device for image and direction input.
How:
- The M5 Core S3 contains a camera and IMU. It will be used as a peripheral device to gather data for processing by the Jetson Nano.
- The M5 Core S3 will compress images to JPEG format and transmit over a USART connection to the Jetson Nano.
- The M5 Core S3 will send IMU data in JSON format via USART to the Jetson Nano.

7.1.1.3 Laser Module

Function: Shoots lasers at opponents' targets.
How:
- The Laser Module will receive PWM commands from the Jetson Nano to position the two servos.
- The Laser Module will receive a command from the Jetson Nano to fire the laser.

7.1.1.4 Shield System

Function: Protects the robot from incoming laser shots.
How:
- The shield system will receive PWM commands from the Jetson Nano to direct the servo.
- The shield system will have an active time limit and a cooldown period controlled by software timers.

7.1.1.5 Lidar System

Function: Scans the environment for navigation and obstacle detection.
How:
- The Lidar system will scan continuously, providing environmental data to the Jetson Nano via a restful web API.

7.2 Software and Network System Requirements

7.1.1.6 Restful Web Services for Lidar Communication

Function: Provides real-time positional data from the Lidar base station to robots.
How:
- A RESTful API will serve data to robots over the WiFi network, allowing them to query for updated positions.
- Data will be transmitted in JSON format, providing coordinates and positional data that the robots will parse for navigation.

7.1.1.7 Robot Control System

Function: Manages robot movement, laser firing, and shield deployment.
How:
- The Jetson Nano will control motors for movement and handle laser firing and shield deployment based on game events.
- Laser hits, shield activation, and robot disabling will be managed via inputs from sensors and communicated through GPIO pins.

7.1.1.8 Image Processing and Obstacle Avoidance

Function: Detects and avoids obstacles using camera input.
How:

- The Jetson Nano will process video input and run image detection algorithms to identify objects and obstacles.
- It will communicate obstacle data to the neural network to adjust the robot's path accordingly.

7.3 Functional Game Mechanism Requirements

7.1.1.9 Laser Hit Detection

Function: Registers hits when a robot is struck by an opponent's laser.
How:

- A sensor placed on the robot's target will detect laser hits.
- The control unit will disable the robot for 3 seconds upon detecting a hit, preventing movement or laser firing during this time.

7.1.1.10 Shield Activation and Cooldown

Function: Deploys the shield to block incoming lasers.
How:

- A servo-controlled shield will activate for a preset duration, blocking lasers.
- The system will enforce a cooldown period during which the shield cannot be redeployed, preventing constant protection.

7.1.1.11 Communication via WiFi

Function: Synchronizes data between robots and the Lidar base.
How:

- Robots will exchange data over WiFi, including position and game state, using a low-latency protocol like MQTT or HTTP.

8 Glossary of Terms & Works Cited

8.1 Glossary

Autonomous Robot: A machine capable of performing tasks without human intervention by using sensors, AI, and control systems.

Laser Tag: A competitive game where players (or robots) score points by "tagging" each other with infrared beams.

LIDAR: Light Detection and Ranging; a sensor that measures distance by illuminating a target with laser light and measuring the reflection to create a map of the surroundings.

Machine Vision: Technology used by robots to process visual information and identify objects or environments, often using cameras and image recognition algorithms.

SLAM: Simultaneous Localization and Mapping; an algorithm used to create a map of an unknown environment while simultaneously tracking the robot's position within it.

IMU (Inertial Measurement Unit): A device that measures acceleration and rotation to help determine the orientation and movement of the robot.

Jetson Nano: A small, powerful computer designed for AI and machine learning tasks, often used in robotics for real-time processing of data like images.

M5 Core S3: A microcontroller unit used to interface sensors and cameras, providing data to the robot's main computer for processing.

PWM (Pulse Width Modulation): A technique used to control the speed of motors and the position of servos by adjusting the duration of pulses in an electrical signal.

Neural Network: A computational model designed to simulate the way a human brain processes information, used in machine learning to predict actions based on input data.

RESTful API: Representational State Transfer API; a web service that allows robots to access and share real-time data, such as LIDAR positioning, over a network.

Obstacle Avoidance: The capability of a robot to detect and navigate around obstacles in its environment.

Servo Motor: A motor that allows for precise control of angular position, used for tasks such as moving a robot’s shield or laser.

Hit Detection: The system that registers when the robot has been "hit" by a laser, triggering a response like disabling movement temporarily.

Database Logging: The process of recording events, such as laser shots or hits, into a database for future analysis.

Disruptor Role: A specialized role for a robot that focuses on obstructing other robots during the game, preventing them from scoring points.

Raspberry Pi: A small, affordable computer used for programming and controlling various robotic functions, including the Disruptor role.

8.2 Works Cited

OpenAI. ChatGPT. OpenAI, 2023, https://chat.openai.com/

End