Beaters 1
Eric Jiang, Michelle Tai
TA Amanda
ECE 110 Section 5
20 February 2019

Conceptual Design Report for Quidditch IDC

We have adhered to the Duke Community Standard in completing this assignment.
Eric Jiang and Michelle Tai

Introduction

Our team's role is Beater #1 in the 2019 IDC, a Quidditch game in which 5 BOEbots assume the roles of 2 beaters, 2 chasers, and 1 seeker, and must move along individual lines of play featuring 5 hashmarks. The BOEbots must be fully autonomous and capable of relaying data to/from each other in order to successfully move about the pitch and score points. When the seeker "catches" the snitch, the bots all must display the total number of points earned in order to officially win the match. A chaser can only "score" when the following conditions are satisfied: it detects a quaffle at a hashmark in its line of play *and* its beater partner, which is our team, has simultaneously "blocked" a bludger at the corresponding hashmark by detecting the object at said location. The seeker must "count" the number of mirrors on its line of play, go to the corresponding location on the pitch, and scores 150 points if it catches the snitch.

Planning and Management

The Gantt Chart outlines major due dates and tasks/steps that our group will take to complete the major challenges and deadlines. We based our Gantt Chart on the lab schedule, which lists which days there will be IDC checks. Between IDC Demos, we will complete the physical building and programming tasks at least 2 days before the next IDC check so that we will have enough time to test our bot and fix any malfunctions. We also split the subtasks evenly between the two group members; however, we will collaborate and work together for most of the time. The task will longest is "Integrated Sensing," which requires coordinating our bot's movement, sensing, and communication so that they occur simultaneously. This is expected to take around 8 days to complete. The tasks that will take the shortest amount of time are the tests for each IDC. We expect this to take the least amount of time since our testing builds off of each other, so the success of previous tests will hopefully contribute to smoother tests later on in the project.

Trade Study/Pugh Matrix

1. Problem: Design a BOEbot Beater #1 that autonomously patrols a line of play to determine the presence/absence of an RFID tag at each of the 5 hashmarks. It must then communicate to Chaser #1 that the bludger(s) at the corresponding locations have been blocked.
2. Alternatives:
   a. RFID Card Reader Module
   b. ColorPAL module
   c. QTI Sensor
   d. PIR motion sensor
3. Trade factors:
   a. Cost
   b. Accuracy
   c. Code complexity
   d. Detection latency
   e. Circuit Complexity

Table 1: Tabulated Specifications for Each Alternative

	Cost ($)	**Accuracy (%)**	**Detection Latency (ms)**	**Circuit Complexity (#wires + resistors)**	**Code Complexity**
RFID Card Reader Module	49.99	90%	~500	5	Minimal
ColorPAL module	29.99	80%	~800	3	Some
QTI Sensor	9.99	90%	~700	6	Moderate
PIR Motion Sensor	14.99	70%	~1000	4	Extreme

These 5 factors were chosen due to their potential impact on the success of Beater #1. In order to win the IDC, Beater #1 must be both quick and accurate when it comes to detecting RFID tags. In addition, in order to address concerns with respect to reproducibility and debugging, the circuit/code must be relatively easy to navigate. Finally, as money is a finite resource, the cost of the sensor must be taken into account when considering alternatives.

Table 2: Normalized Quantization Scale for Factors

Normalized Value	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Cost ($)	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
Accuracy (%)	95-100	90-95	85-90	80-85	75-80	70-75	65-70	60-65	55-60	50-55
Detection Latency (ms)	50-150	150-250	250-350	350-450	450-550	550-650	650-750	750-850	850-950	950-1050
Circuit Complexity (#wires + resistors)	3	4	5	6	7	8	9	10	11	12
Code Complexity	Almost nothing	Extremely uncomplex	Minimal	Relatively uncomplex	Some	Moderate	Good bit	Very much	Extreme	Indecipherable

Table 3: Weighting Factor/Rationale for Factors

	Multiplication factor	**Rationale**
Cost ($)	2	Since we will not be purchasing multiple units of each sensor, we place less importance on cost and more weight on functionality; we would prefer to use a more expensive sensor that works as opposed to a cheaper one that does not.
Accuracy (%)	8	In order to successfully complete our job as Beater #1, we must be able to accurately and consistently detect an RFID tags on the hashmarks. It is vital that the chosen alternative is accurate (low frequency of both false positives and false negatives), such that the maximum amount of points can be scored by Chaser #1.
Detection Latency (ms)	5	Beater #1 must not only be accurate in its detection of RFID, but also quick to do so. After all, the faster it can recognize the RFID, the faster points can be scored. Thus, it is important that the detection latency is as small as possible.
Circuit Complexity (#wires + resistors)	1	Each of the 4 proposed alternatives feature sensors that were specifically designed to interface with the PARALLAX board; thus, all should be relatively simple to implement in terms of circuit complexity.
Code Complexity	4	The more complex the Arduino code is, the more difficult it is to follow. This can result in more error-prone behavior that is difficult to efficiently debug/fix.

Table 4: Total Value Tabulations for Factors

		RFID Card Reader Module		ColorPAL Module		QTI Sensor		PIR Motion Sensor	
	Weight Factor	**Norm. Value**	**Total**	**Norm. Value**	**Total**	**Norm. Value**	**Total**	**Norm. Value**	**Total**
Cost ($)	2	6	12	8	16	10	20	9	18
Accuracy (%)	8	9	72	7	56	9	72	5	40
Detection Latency (ms)	5	6	30	3	15	4	20	1	5
Circuit Complexity (#wires + resistors)	1	8	8	10	10	7	7	9	9
Code Complexity	4	8	32	6	24	5	20	2	8
GRAND TOTAL			**154**		121		139		80

Discussion

Utilizing the Trade Study method, the RFID Card Reader Module was selected as the most viable alternative. Since it scored the the highest of all alternatives on the 3 most highly weighted factors (Accuracy, Detection Latency, and Code Complexity), it follows logically that the RFID-based alternative should be pursued by the team. While slightly expensive, the RFID module is the clear winner with the highest potential to succeed during the IDC.

Cost Estimate

Table 5: Costs of BOEbot components

Component	Cost ($)	Quantity	Total Cost ($)
Arduino ATMEGA 2560	51.91	1	51.91
BOE Shield For Arduino	39.99	1	39.99
BOE-Bot Aluminum chassis	24.99	1	24.99
BOE-Bot Plastic Wheel w/ Tire	3.99	2	7.98
BOE-Bot Tail Ball Wheel	3.95	1	3.95
BOE-Bot Li Ion Power Pack With Cable and Barrel Plug	49.99	1	49.99
Red LED	.32	1	.32
Green LED	.32	1	.32
220 ohm Resistors	.10	3	.20
Li Ion Cell	8.99	2	17.98
2A Fuse	1.12	1	1.12
Continuous Rotation Servomotor	13.99	2	27.98
USB A to B Cable	4.99	1	4.99
7.5v 1A Power Supply	14.99	1	14.99
XBee SIP Adapter	34.99	1	34.99
2x16 Serial LCD	29.99	1	29.99
RFID Sensor	29.95	1	29.95
Push Button	.50	1	.50

The total cost for the robot so far is **$342.24**.

Confirmation Emails:

Thank you for submitting your entry. A copy is included below for your records.

ECE 110 Inventory

Item	RFID Reader
Quantity Requested	1
Bot Box	47
Name	Michelle Tai
NetID	mrt36

Thank you for submitting your entry. A copy is included below for your records.

ECE 110 Inventory

Item	Serial LCD
Quantity Requested	1
Bot Box	47
Name	Michelle Tai
NetID	mrt36