### Robot Challenge Journal

Challenge No.: 3

 Team Name: Team E

Members’ Names: Carlo Consolacion, Eric Nelson, Cody Shafer, Jesse Kitteman

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Seq. #** | **Name(s)** | **Hypothesis/Behavior** | **Description/Results** |
| 5/7 | 001 | Carlo,  Eric,  Cody,  Jesse | Began planning and organizing the group work and options and the added diagrams for behaviors. | Defined the different behaviors and put them in a Collection Task Behavior Network, a Finite State Machine or different sensor behaviors, and a hierarchy of the different behaviors based upon hunger. Then we started planning out which bricks would take care what behavior, still have to define Bluetooth communication work. |
| 5/7 | 002 | Carlo,  Eric,  Cody,  Jesse | After the initial planning and state definitions, we began splitting up which brick would have what sensors and then the design of the robot. | We decided to have brick 1 take control of the motors and the behaviors we already know (Approach, Wander, and Line Following) and brick 2 would have the other behaviors (Collision and Escape). Then began the physical design of the robot, possibly having both brains stacked on top of each other, or parallel to each other. Final design was to put the brains sideways and facing outwards (back to back) with the motors easily removed to make battery replacement easy. Then started with the energy scale planning with timers and value scale. |
| 5/7 | 003 | Carlo,  Eric,  Cody,  Jesse | Started doing the energy scale, the timers in the code work with long values; therefore, we went with the long scale to plan how the energy is determined. | From full to dead is a total of 4 minutes, 2 minutes from full to hungry, 1 from hungry to danger, and 1 from danger to dead. Then we started writing pseudocode for the timers and energy level. Making we are always in the same state and decrement things properly. |
| 5/7 | 004 | Carlo,  Eric,  Cody,  Jesse | After talking about how to implement the different states in code for energy levels, we decided to hold off until Monday and sort out the plan for then. | We got bogged down in different issues in pseudocode for the energy level and class time ended, then we decided for Monday to build the robot and get the Bluetooth communication working. |
| 5/12 | 005 | Carlo,  Eric,  Cody,  Jesse | After the weekend, we picked up where we left off. Cody and Jesse started working on Bluetooth and Carlo and Eric started building the robot how we planned. | Rented 2 new bricks to test bluetooth while the physical design is implemented. We are using a slave and master type set up and have started testing by sending and receiving simple messages. Ran into issues of the slave changing a string and then the master receiving it back. Working and testing with it still. The robot base is coming together as well. |
| 5/12 | 006 | Carlo, Eric, Cody, Jesse | When pairing bricks with bluetooth, the connection must be reestablished after a new download. | Once we figured out the connection was broken, our sending messages test worked properly. And the robot is nearly built. It has 2 motors and a ball (possibly), with the motors back to back, each facing outwards. |
| 5/12 | 007 | Carlo,  Eric,  Cody,  Jesse | With our message passing completed, we started testing the other types of messages: booleans, nums, etc. | Made some progress with the Bluetooth, and started playing with sending and receiving using the general methods (which operate with strings) and tried putting boolean values. This failed. |
| 5/14 | 008 | Eric,  Cody,  Jesse | Carlo was out sick for the day. We continued to work on implementation of behaviors by trying a priority based arbitrator. Put the robot together and began getting the old code implemented first. | We started by getting the basic functionality merged (all the code we already have for past behaviors) and decided to worry about Bluetooth afterwards. |
| 5/14 | 009 | Eric,  Cody,  Jesse | Had to redesign the robot a little bit. Added the front bumpers for reactions, and the light sensors were moved onto the new bar in the front instead of mounted to the top sonar sensor. | The original bumper design never detected if both sensors hit, so we redesigned it like Jesse’s challenge 1 robot bumper and connected light sensors to that because the original bar that held the light sensors was too high and ran into the other bar. The robot seems slightly front heavy but other than that, everything else seems good. |
| 5/14 | 010 | Eric,  Cody,  Jesse | Updated our Arbitration diagram and had Fowler check up on it to make sure we aren’t going in an impossible direction. | Our state diagram looks good and doable according the Fowler, and so then we planned to continue as we were with our different behaviors. The robot is also fully complete as well. Now for the actual implementation of the code. |
| 5/19 | 011 | Eric,  Cody,  Carlo, Jesse | Over the weekend break, Eric began coding at home for an arbitrator similar to the one we planned on. Carlo returned back at full health today. | Eric began testing his arbitrator before class while the rest of us began starting on different parts of the project. Carlo wanted to take the hunger meter using Cody’s scale from last Thursday (120 for meter and decrementing by 1 every 2 seconds and increase by 2 every 2 seconds to give the proper amount of recharge and drain time using whole values (ints instead of doubles). |
| 5/19 | 012 | Eric,  Cody,  Carlo,  Jesse | Eric is currently working on the arbitrator. The implementation is to run a task (thread of the behavior) and stop lower priority tasks from running by stopping their threads or having them yield | Using a trivial example, we tested the yield method to test task priority and switching (playing sounds) and then yielding when its not supposed to be running. The test worked and initial theory works; therefore, we began to implement and test it to see if we can make it work. |
| 5/19 | 013 | Eric, Cody,  Carlo,  Jesse | Simultaneously to Eric, Cody and Jesse are working on the escape keeping the journal updated. Carlo is working back and forth with Eric while trying to get a basic test for his energy meter. Meshing everything together once each individual part is working will be a challenge. | Cody got a full test for getting the Escape behavior to work (which is on the Master Bluetooth brick (different from the Controller brick)). The only modification will be adding the message passing and the collision code (we already have collision). After some more thorough testing, we were able to get the individual Escape behavior to work, and added the Bluetooth functionality. |
| 5/19 | 014 | Eric, Cody,  Carlo,  Jesse | As everyone progressed, some functionality that was new started to come together (mainly the Escape). Then we started figuring out how to send the bool values through Bluetooth to show that the bricks are communicating. | With the code for the Escape complete, we began to start the Bluetooth part as well as plan out the gradient and line following. Carlo had the idea of using one light sensor to do the line following but there were different ideas for each method depending on the light sensors. |
| 5/19 | 015 | Eric,  Cody,  Jesse,  Carlo | Using the API for Bluetooth, our initial design was to keep the message passing to a minimum and use a simple slave and master technique. The Lego bricks talk by checking mailboxes with left messages from the other one. | Our initial implementation was approved early on in the behavior hierarchy. Master Bluetooth bricks always initiate communication. The secondary control brick will be our master Bluetooth brick and send flags for when sensors are triggered (Escape and Bumpers). Our implementation has started, but has not actually been tested yet. We will get to this the next class time. |
| 5/21 | 016 | Eric, Jesse,  Cody,  Carlo | Continuing where we left off, we decided to implement every behavior except for the gradient and line following. This includes the message passing for the Bluetooth communication. Carlo is working on a way to use our 2 stationary sensors for the line following and patch feeding. | Our minimum goal is to get the master control brick to have all of its functionality except the gradient following (Approach and Wander). And afterwards get the Bluetooth working for collision and escape behaviors. Initial escape code was done and completed, just needs to be merged with collision on the master Bluetooth brick. We have re-usable code that does approach and wander properly already, implementing the Bluetooth is the difficult part. Using the mailboxes from the API, we plan on sending boolean flags to keep the communication dependency to a minimum. |
| 5/21 | 017 | Eric,  Jesse,  Cody,  Carlo | In an attempts to add Bluetooth communication, we have all of the initial code set up for the behaviors and message passing, its just a matter of testing at the point. And we figured out the line following problem in theory. | While working through the line following problem, we learned that as long as we don’t hit the end of the line, its ok to turn around. We were trying to have it turn around almost immediately. This made the problem semi easier to solve. Our logic is that if both sensors are on, feed; if one goes off, then correct back the other way to get both on. If the other goes off, then you know your line it getting smaller so turn around, but if both go back on, continue feeding. If both go off, back up and continue feeding until done. Carlo and Jesse are modifying the challenge 2 code |
| 5/21 | 018 | Eric, Jesse,  Cody,  Carlo | Eric and Cody have been working on the master control brick. It now has wander and is currently being updated with the approach behavior. | The robot now successfully wanders around and will soon approach objects the same way from challenge two. After that we need to implement the code to run each of the other behaviors because the Bluetooth will be sending flags over. |
| 5/28 | 019 | Eric,  Jesse,  Cody, Carlo | After the long 3 day weekend, we dove right back into getting the project done, since we only have 1 week left before the break. Carlo and Jesse are working on the gradient following, Eric and Cody continue work on a new arbitrator. | After the break, minimal work was done; however, we planned out the next week to meet up on Friday and implement more behaviors to attempt to finish over the weekend. Most behaviors are implemented except for a complete gradient following behavior, its almost complete, just some more fine tuning. Same with the arbitrator. It has a problem switching states properly. |
| 5/28 | 020 | Eric,  Jesse,  Cody,  Carlo | Redid the state diagrams and the behavior diagram in UMLET and made it look legible. Had it rechecked by Fowler. | After redoing the behavior and finite state machine diagrams in an electronic format. Jesse ran it by Fowler again and this time it fit the criteria. |
| 5/28 | 021 | Eric,  Jesse,  Cody,  Carlo | Upon getting the gradient following tested, we had a special case that when entering a patch from a tendril, it wouldn’t properly turn around and follow the line. But this was fixed. | After multiple classes of testing, the battery was pretty drained. After a battery replacement, the motors were much faster. Carlo changed his speed values, and it seemed to work well. After the battery change, his 180 turn then turned too much because of the power of the motors. Slight tweaking of this made it properly follow a gradient. |
| 5/30 | 022 | Cody | Cody reworked the arbitrator, tested reading in from Bluetooth, and tested most of the combined code. | Cody got a working arbitrator that properly switches states, with a small minor bug that approach behavior takes over. There were some bugs in the message passing when testing the collision, but the escape message passing still worked. Ran into errors with the getting the collision to work properly. |
| 5/30 | 023 | Cody, Jesse | Jesse joined Cody after work to help make progress on the robot. The current issue was the collision reactions. | Cody took a break, and Jesse played with some of code and changed the value of the bumper check. Then tweaking the turn subroutines seemed to fix the issue. Testing the robot on the ground proved he made the proper turns and reactions. |
| 5/30 | 024 | Cody,  Jesse | After the bumpers were working, we then added back the approach behaviors back to test the behavior hierarchy, which worked well. And then we made sure that collision was a ballistic behavior. Then finally we tested the escape reaction. | With collision working, we tested the arbitration between behaviors and ballistic collision behavior. Both of these were successful; however, the escape routine was not as such. We encountered several issues, infinite looping, switch states but not actually reacting, not adjusting to consecutive flash, and not switching into the state. After some tweaking and reworking, everything was fixed. One major issue was that we would divide by zero instead of one for the first flash. And then the issue of reacting more than four times came up; this issue was resolved with a simple if check statement. |
| 5/30 | 025 | Cody, Jesse | Once we had the initial code working, we then moved on to getting the gradient and patch feeding implemented. Carlo had the code in a separate file that ran under the main thread with no subs or anything. Our remaining objectives for behaviors is to implement these last too and fully test all of the behaviors. | When trying to merge the last bit of remaining code, we tried to avoid rewriting the whole task to make it more easily implemented since we tested that it worked. We found a StopTask method on the BricxCC that would have worked; unfortunately, we do not have the latest firmware compatible with this method. |
| 5/30 | 026 | Cody,  Jesse | After we implemented gradient following and other code changes and bug fixes. Nothing seemed to work well together. Collision only worked when we were in the escape state. Approach would take over and hold the state . But collision quit working altogether. | We thought we were in the final stretch; however, upon testing everything seemed to fail and not work. Our states quit working and changing properly. Bluetooth message passing worked fine still. Possibly an arbitrator issue. We gave up after hours of debugging and little progress. We attempted one last final attempt to get something to work. We took the code where we forked off into a non-full branch and tried to run what originally worked. We had the same issues with the old code. Still have no idea what the issue is. |
| 6/2 | 027 | Cody,  Eric,  Jesse | After working on the project over the weekend, we dug in and picked up by trying to fix our basic behaviors. With more then 2 people working on it, we should be able to fix the problem relatively quickly. | As an attempt to fix the collision problem, we moved the bumper bar out a little bit to ensure that the touch sensors were always being hit. Initial testing of the bar seemed to help, even if it was just a little bit. After a little bit of code manipulation, we were able to pinpoint the problem being somewhere in the escape state. When removing the task from the list (in the precede method), it caused all states to switch properly. |
| 6/2 | 028 | Cody,  Eric,  Jesse | We cleaned up the code in the escape behavior once we were able to figure out that escape was the issue. Next we ran into the issue that while escaping and then receiving a bumper collision, the robot would just stop, and freeze. | Now that we made it back to where we were on Friday. We continued testing. A bug surfaced where we got stuck in state 6. After testing, we couldn’t recreate the bug, so for now we ignored it and moved on. |
| 6/2 | 029 | Cody,  Eric, Jesse, Carlo | Status Report: Bluetooth communication is working, escape, collision, wander, gradient/feeding (in a separate file still).  Need to Finish: Approach (Proportional), code merge with all of the behaviors, energy level, and timers. | We are currently working on getting the Approach behavior to work properly now that the arbitrator has been fixed. In addition, we are fixing the gradient and patch feeding code so it is an easier is implementation to copy and paste. |
| 6/2 | 030 | Cody, Eric, Jesse,  Carlo | Approach is mostly implemented now. Just needs a little bit of fine tuning. We switched the placement of the upward light sensor and the sonar to ensure that no excess noise would get in the way of the sensor. | Carlo and Jesse were able to get the gradient following and patch feeding implemented into an older but working version of the arbitrator (missing the fine tuning of a couple values). There is a small error where the robot won’t default to wander with the gradient task. It just needs to be reworked a little to implement all the features of the arbitrator: death, defaulting to state 1, and proper switching of states based on the hierarchy |
| 6/2 | 031 | Eric,  Carlo,  Jesse | After the initial class time ended, Cody left for his other class and the rest of us kept working. Much progress was made. The gradient is implemented and works with the arbitrator code and the approach seems to work well, too, after a little tweaking. The wander was a bit choppy so Eric rewrote that algorithm and it works much better now. | The Approach works proportionally and smoothly now. Before it was quite choppy and would get too close to objects. In addition the wander behavior is more smooth and less random, giving a more smooth drive. Carlo started to implement the feeding and hunger states and Eric began cleaning up code while Jesse helped each of them and documented the journal. |
| 6/2 | 032 | Eric,  Carlo,  Jesse | The energy level has been slowly tested. The initial trials (10 seconds between full and dead). The death state is implemented as well. | The energy level runs for 10 seconds until dead. And then feeding initiates after 5 seconds. This was all testing and worked; however, a bug was found that when feeding, you never stop feeding so it never goes back to wandering other than on the patch in circles. |
| 6/2 | 033 | Carlo,  Jesse | After some more progress, Eric had to leave. Carlo and I continued working and tested the hunger and switching of the states more thoroughly. | After testing the initial states we implemented sound for the switching between different energy states. All the proper states were ignored for danger and it fed properly on a patch and detected a gradient. Everything up to this point seemed to work in unison except for the approach. We decided to call it quits after this was complete. |
| 6/3 | 034 | Cody | Cody came earlier in the morning to work on the approach. The primary goal was to fix approach and then test for complete functionality. | While fixing the approach, Cody spent time reworking the code for approach and testing it. After a few hours, it worked by itself in testing (with no other tasks being ran). Upon merging all the tasks together a new bug surfaced. |
| 6/3 | 035 | Cody | While testing a rewritten approach, the wander and approach code collided. The arbitrator constantly would switch back and forth between the two different tasks, and the behaviors would result in choppy behaviors. | The behavior states would constantly switch between wander and approach. This was fixed by moving the state reset back to wandering in an if-else block instead of at the end of the task. This fixed all the bugs with wander and approach fighting for control. |
| 6/3 | 036 | Cody,  Eric | After the approach and wander worked with each other, wander would randomly stop moving and cause a stand still. | The final bug was the wander would randomly stop moving the motors but would still react to other inputs. We found a break; statement in the middle of wander, and it was supposed to be a continue. Once that single line was changed, we thoroughly tested everything and it all worked in unison. |
| 6/3 | 037 | Cody,  Eric | After everything was working, Cody and Eric wanted to rework the escape behavior. Any changes they made seemed to not work well and cause more issues than fixed. | Eric and Cody tried messing with values and refactoring the escape task. The habituation seemed to scale faster than we originally thought, so it was difficult to tell after the second flash but the reaction worked. The final decision was to just leave it as is since it worked and to get feedback from the first demo before changing anything. |
| 6/3 | 038 | Cody, Jesse | Challenge 3 Demo 1 | We used our first demo after all of the behaviors seemed to work well. The demo ended up being 100%. It was able to ignore patches when full and react to bumpers above all else and die after 4 minutes. We were the first group to properly implement the gradient following though. The robot detected the line was getting smaller before the end of the line and turned around and continued feeding. The arbitrator switched behaviors properly and the food incrementing increased twice as fast as the decrease. Demo 1 concluded our code. |