Gaga Robots

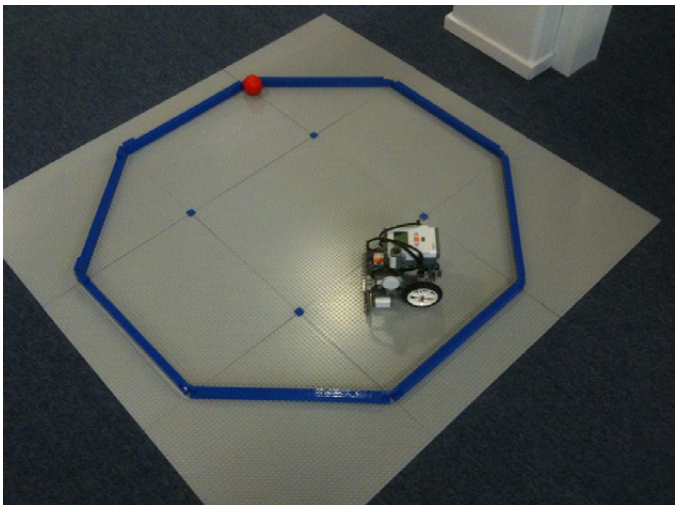
By 1brick

Gaga is a form of dodge ball I found at a summer camp where I used to teach a full on 8 week hardcore LEGO® MINDSTORMS program and was one of THE games to be good at camp. It was the true test of camphood, where bloody knuckles and scraped knees were badges of honor. The game was played in an octagonal arena about 5 meters by 5 meters, with a concrete floor, 4 foot tall walls and a door to let combatants in and out as they were struck. The ball was a hard, but springy kickball, forgiving for the game, but hard on the face if you were unlucky enough to get hit there. Your target was everything below the knees, but you were not allowed to pick up the ball, only strike it with your fist or hand. You weren't able to hit the ball more than once unless it hit another surface first, such as the wall, or another person's target area and hitting the ball out of the gaga ring would get you out of the game. The game is epic, and everyone plays as hard as they can. The kids (and counselors) had been begging me to build the game out of LEGO the whole time I worked there.

At first it would just be a model, but then maybe, just maybe a working MINDSTORMS game. I never had the time (or the resources) until I came to my current job, even though it would be just for a year. I finally had an opportunity and a reason when an event called BEEP rolled into town, and the MINDSTORMS booth was looking for something new and novel to bring. So about a month or so before the event, I decided to take it on.

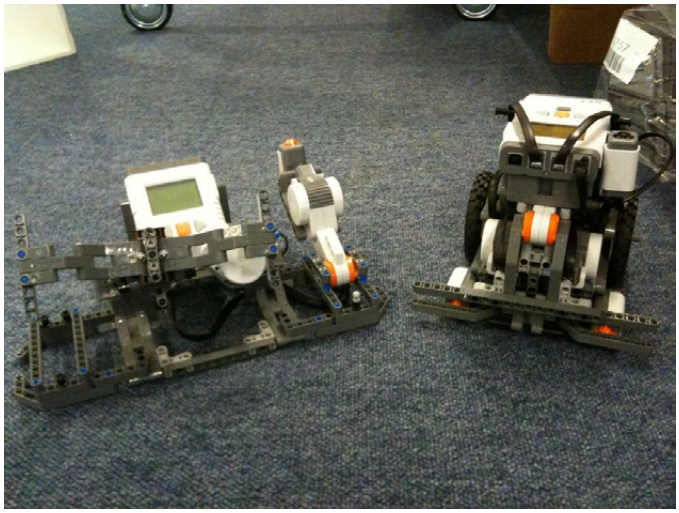
The ring was what really made the game interesting. It felt like standing in a gladiators arena. The original eight walls of the game were held up by thick wood pillars and the walls themselves were old, beaten and worn but strong and full of character. The LEGO version had to feel just as epic. What started out as blue brick borders on top of 9 grey baseplates became a fearsome black wall on top of a custom (sort of) laminated wood base with mitered corners. The 16 high walls were light but strong. Internally they have technic brick frame and the walls are 8x16 tiles on both sides adding strength and flexibility. They were held to the base with Dual-Lock, commonly used on FLL Tables. I was able to secure the wood base through one of the project managers where I work. He had it custom made just for me! Having the right ball was pretty important too. The Duplo ball could do the trick, but it was hard to get enough inertia the from the robot's puncher. I tried some other larger balls as well, rubber and foam with different weights. The best solution came to be multiple balls. At some point the game had been played with over 15 Duplo balls, and it makes the game that much more exciting.

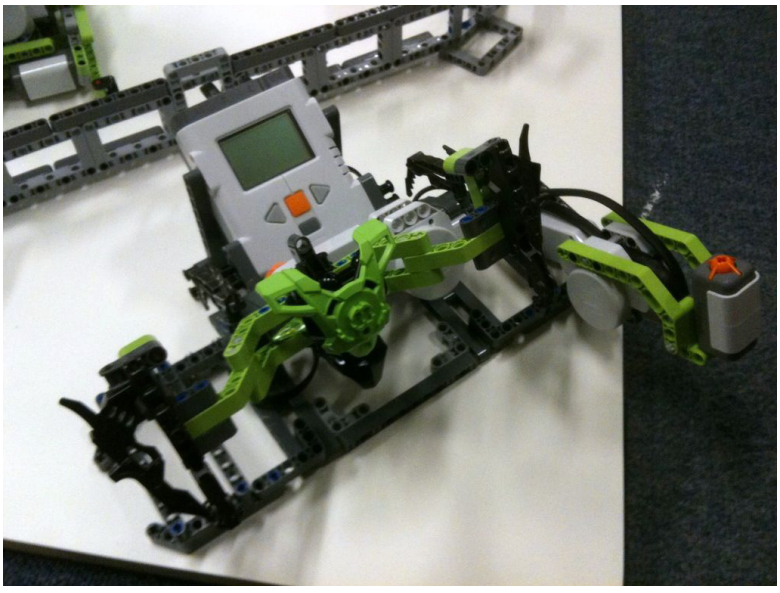
So how to build a robot that can be hit by a ball, and know that its been hit? I had a couple of goals with the robot. First that it



would be functional as a dodgeball playing robot and that it be simple enough to duplicate, not just by me, but maybe a child who might want to build it. The robot had to be able to detect touches or hits from the ball, so an array of touch sensors were definitely needed. Two in the front and one in the back, both as low as possible so the ball can strike from rolling on the ground. Each sensor also had to get a surface area big enough that could be struck and trigger a response.
The rear one was just a little easier than the front two. Those needed a spring type mechanism to bring the double bent beam triggers back to place. A rubber band solution fixed that. Getting the puncher in the right place needed an interesting solution as well. The effector had to take the same place as the touch sensors but not interfere with them. I had to build it wide enough to be able to hit a ball with strategically placed gaps for the touch sensors. Eventually I added a color sensor so that the user could easily see how many 'lives' the robot had left.

The controllers were based on a steering wheel I designed a few years ago. A single lever controlled throttle while the wheel did steering functions. A touch sensor was added as a trigger for the puncher of the robot. On this version of the controller, I had the NXT screen visible to show some status of the robot while its running. Ideally it should show the number of lives you had left at the very least, and maybe show which sensor was hit as well. The original controller had two additional touch sensors for 'paddle shifting' speed of the target robot, as well as a light indicator which indicated speed by how bright the light was. I tried my best to have some sort of controller that was visually appealing as well as functional and comfortable to use. Though the steering wheel is mainly used with one hand, it's more like a racing steering wheel that's used for two. The grip of the wheel should be comfortable to most kids, but not so comfortable that they'd want to stay on it for hours on end. I used some Hero Factory parts to give the wheel some styling and to get the right shapes. The HF chest part was just perfect and came in all the right colors for the best finish. There are some clever tricks in getting that chest in the right place, but I think it came out alright. Wiring was the trickiest part in building the controllers, as its pretty tight and you have to be able to keep all the moving parts as loose as possible.





My initial prototypes were built in dark grey, that way I knew I had most of the available parts. The goal was also to have this robot be accessible to those who had the MINDSTORMS Education set 9797 and the Resource set 9648 or 9695. I believe its still possible, but there might be some adjustments. For the full set, each robot and controller were done in four colors. Now, every color doesn't have every part, especially lime green. Making the right color substitutions would make an impact on the look of robots. For the controllers, only the joystick and the steering wheel were colored and this gave just the right effect. It also made building that much faster.

Programming wasn't so bad for the robot itself. The meat of it all was the idea that all three touch sensors had to react and lead to the same common end result- that the robot was hit! While doing this in language programming is easy, it just looks complicated in NXTG. By using a series of logic blocks in the OR function, each sensor was easily daisy chained to the others.

Controlling robots via Bluetooth had been so many times before, so that was pretty much already written for me. The fun part here was to connect the robots automatically and also have a clever way of having the robots 'die' when their lives were up. I used a counter to count down lives when a touch sensor was struck and as the lives went down to certain thresholds, the color sensor would display a different color. Green for starters, Blue in the mid

range of lives and red as you were about to die. The robot would flash red for five seconds before completely stopping. This allowed the driver to get the robot back to himself to reset the robot manually. In the end each controller had to have its own program as each had to have a unique ID to the robot, however, each robot could have the same program.

In a perfect world, RoboGaga would play just the same as the real thing. You would punch the ball and try to hit everyone else where they are vulnerable, and if they ran out of lives they would leave the game. Clear winners could be established that way and you can have tournaments. This isnt always the case. At BEEP, ROBOGaga made its debut, I found that teaching rules to casual onlookers was just too difficult, and this was on top of a language barrier as well. Patrons barely understood the concept of driving around and hitting the balls. It was hard to explain rules in a crowd and the best I could do was to teach onlookers how to drive the robots. Though the game didn't play as expected, it was still great fun for the spectators and even more fun for those who played. One day though, I hope to run a real game.

The next stop for RoboGAGA will be at LEGOWorld Copenhagen. Lets see how those kids do with the game
#