Whac-a-Mole

Things That Think
Spring 2003

Stephany Bader
Heather Buna
Ryan King

Abstract

Our mechanical automaton is modeled after the ever-popular arcade game *Whac-a-Mole* (1970). The original game has a target surface with five holes through which animated moles pop in and out of at random. The player must hit as many moles as possible with a mallet. If the player successfully hits a predetermined number of moles, he advances to the next level where the moles are even more clever and elusive as they pop in and out of their holes faster and faster with each level.

Ours is a miniature version of *Whac-a-Mole*. There are three moles that pop in and out of their respective holes at random. Crickets are used to randomize the motions of the moles, motorize their movement, and notify the player of all successful mole hits.

Design

Each of the three moles is attached to a follower that sits atop a snail cam. The snail cam rotates using a crosshatch axle that is turned with a motor.

Snail Cam

We chose to use a snail cam to guide our moles in and out of their holes. It consists of two parts, a cam and a follower. The snail cam transforms its rotational motion into a vertical upward motion as the follower is lifted to the highest radius of the snail cam. Once the follower reaches the end of the snail cam, it drops down abruptly to the smallest radius. This allows our moles to come up out of their holes, and then quickly drop down into hiding again.

The follower consists of a round wooden dowel with a metal bearing attached to the bottom that rolls along the snail cam. Clear plastic tubing that is fixed in place encases the follower. This allows the follower to easily slide up and down through the tubing while giving it the vertical support it needs to keep from sliding off of the snail cam.

Motor

The motors, powered by 9Volt batteries, are the source of movement for our moles. We chose to work with *Lego* motors since they have small gears already attached to them. One problem we faced in working with the motors was that their speeds are constant and can't be varied. This was a problem because we didn't want our moles popping up and down too fast. Because the speed of the motors couldn't be varied directly, we had to vary them indirectly using gear-ratio techniques. Using a pulley system, we used the small gear powered by the motor to turn a larger gear that rotated the axle of the snail cam, thus slowing down the rotational velocity of the snail cam and the vertical speed of the follower.

Touch Sensor

Attached to the bottom of each mole is a touch sensor. We used *Lego* touch sensors since they provided the easiest way of attaching our moles: we simply used small crosshatch axles to connect the moles to the sensors. As a mole is pushed downward, it triggers its sensor causing its Cricket to beep—an indication that the player successfully whacked the mole.

Cricket

The minds behind our moles are the Crickets. They are what allow the moles to seem so clever and elusive. Each mole is controlled by its own Cricket. The Cricket plays three main roles in our mechanical automaton: it randomizes the exposure of the mole, it tells the motor when to turn on and off, and it provides audible confirmation when a mole is whacked.

Since the rotational velocity of the snail cam depends on the strength of the battery, we initially have to test how long it takes the snail cam to go through one revolution, which corresponds to the mole popping out then back in (one time). Once we know this value, we can program the Cricket to turn the motor on at the required power level for the time required for one revolution of the snail cam. Using a random number generator, the Cricket will turn the motor on once at random during the given time interval, say anywhere from 0 to 8 seconds. Shortening or lengthening this time interval will cause the mole to pop-up more or less frequently, respectively. The Cricket is also programmed to beep when the touch sensor is triggered. So when a player whacks a mole, the sensor sends a signal to the Cricket causing it to beep for audible verification that the player has in fact, whacked-a-mole!

Construction and Pitfalls

We had many different ideas for how we were going to create the up and down motion of the mole. One idea that we almost went with was a piledriver gear. This provided the steady climbing and rapid dropping of the mole. We decided not to go with it mainly for supplies purposes, and because the snail cam provided very similar results.

We knew that it was in our best interest to first make a *Whac-a-Mole* prototype (Figure 1). Except for the snail cam gear, we built our prototype mainly out of *Legos*. The prototype was a very helpful asset to us in making our final model. We were able to analyze different parts of the prototype, decide what needed improvements or change, and what wasn't needed, or was useless to our project.

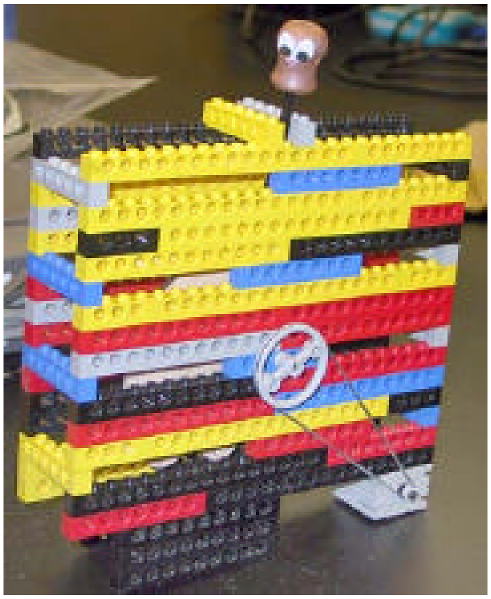

Figure 1: Whac-a-Mole *Prototype*

One of our first realizations from analyzing our prototype was that our snail cam was unnecessarily large. We were able to make a new snail cam that was much smaller, lighter, and did the job just as well.

We also realized that our cam follower might not be as efficient and reliable as we had first hoped. The original cam follower consisted of a wooden dowel, without the plastic tubing and without the bearing on the bottom (Figure 2: Left). It was filed in a triangular fashion to ease its movement along the snail cam, but there was too much friction, and it had a hard time following the snail cam without falling off. So then we revised the follower by cutting a large notch out of the bottom that was large enough to fit over the snail cam (Figure 2: Center). This kept the follower on track, but friction was still causing the follower to get stuck. Then we screwed a metal wheel bearing to the bottom of the dowel (Figure 2: Right). This helped it follow the snail gear tremendously, but there were still stability problems that we needed to address.

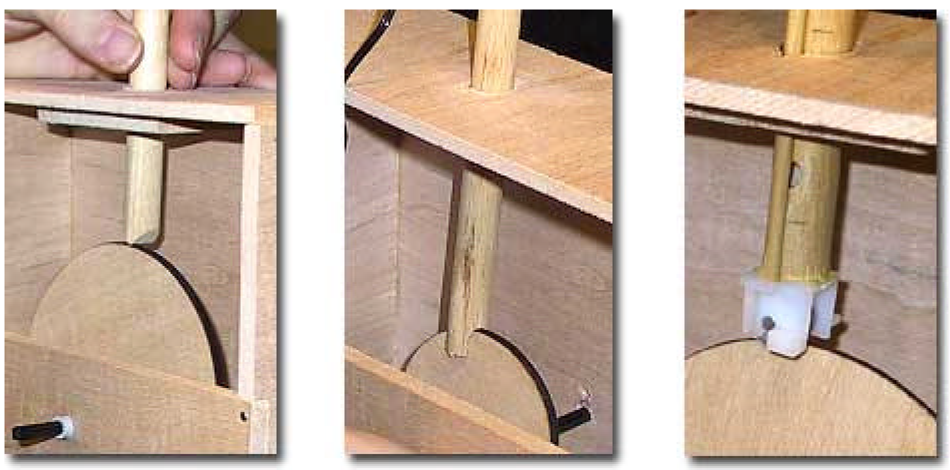

Figure 2: Progress of the Cam Follower

We tried many different techniques to keep the cam follower from falling off of the snail cam. We tried gluing a skinny dowel to the side of the follower to be used a guide. This solved our rotational support problem, but it did nothing for our vertical support problem. It also greatly contributed to the amount of friction, making the motor less efficient. So we revised our follower once more, using the clear plastic tubing as casing for the follower to slide through. This solved both our problems of vertical support and friction simultaneously.

Another part of our design that we seemed to be constantly revising involved the touch sensor and how the mole would trigger it. Initially, we used a conventional touch sensor (not a *Lego* touch sensor). We had the mole sitting on top of the cam follower, which was inserted through a spring that kept the mole at a distance from the touch sensor (Figure 3: Left, Center). When the mole was hit, it caused the spring to contract and allow the bottom of the mole to trigger the sensor. This idea was quickly thrown out: it was hard to line things up correctly, and the spring was so stiff, that it took a really big whack of the mole to trigger the sensor! We wrestled with a few more ideas, one being that we'd use the spring to trigger the sensor, rather than the mole. To do this, we fixed the sensor in place on the top of the cam follower. Then, the spring was attached directly to the mole, and when pressed, the mole would push the spring downward, triggering the sensor (Figure 3: Right). This idea was very flaky and the spring still seemed overly powerful for our needs. This is when we found the *Lego* touch sensors. They were very simple to use and straightforward, and solved our problems perfectly!

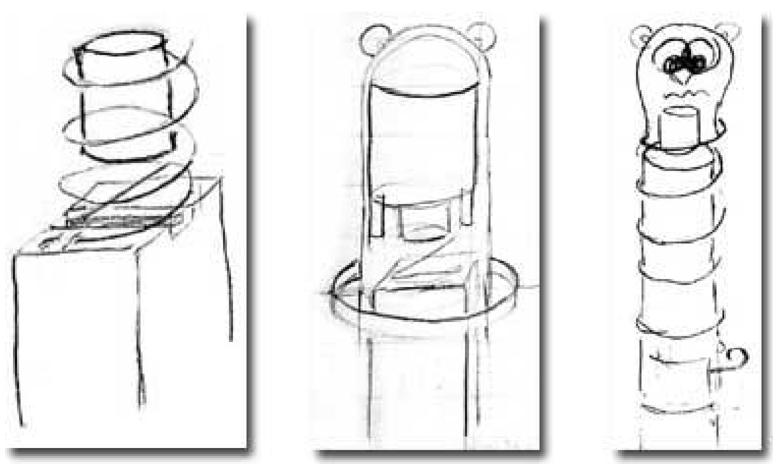

Figure 3: Progress of the Touch Sensor Triggering

Once we had decided on our final model for our automaton, we made two more copies of it for the other two moles (Figure 4). We used the laser cutter to make each of the three snail cams out of balsa wood. Working with this lightweight material allowed the motor to easily rotate the snail cams. The wooden dowel followers were also fairly light, and they didn't pose any problems for the motor as additional weight. The two new mechanisms worked almost flawlessly on the first try, without having to make any major adjustments.

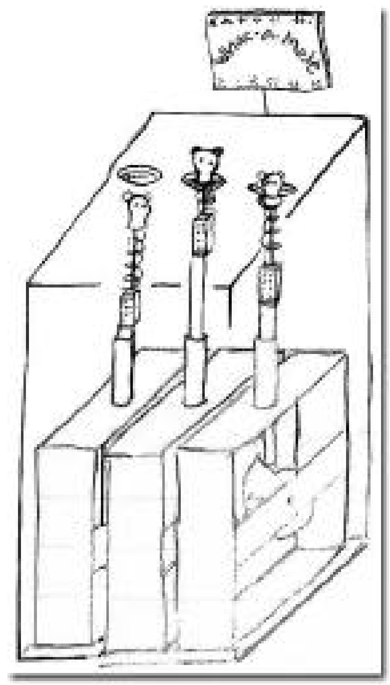

Figure 4: Final Sketch of Project

Evaluation/Education

The physical mechanics of our automaton are fairly straightforward and easy to analyze. When analyzing the *Whac-a-Mole* that we built, the first thing one sees is the mechanics involved in moving the moles. The snail cam demonstrates a beautiful example of the transformation from rotational motion to vertical motion. Another part of our automaton that is easy to see is the rotation of the motor and how it transfers power from its small gear to the larger gear using a pulley system. These basic properties of mechanics are easy to see and understand by watching our automaton in action.

The computational properties of our automaton are more ambiguous to see on the surface. But after watching the automaton in motion for a while, one might realize that the moles come up at random, and thus, one might infer that the Cricket is programmed to enforce such randomization. If one chooses to play with the automaton and whacks a mole, he will surely make the relationship between hitting a mole and hearing the Cricket beep. However, the only real way to know exactly what's going on inside of the Crickets is to analyze the code that we programmed into them.

The original *Whac-a-Mole* has kept its appeal for so many years, possibly because of the anger and frustration a player gets out of his system when pounding on those moles! And although ours is a miniature version of the anger-abating game, if one were to use our automaton as a method of anger therapy, it might be smashed to bits! However, the principles are the same: the moles, though cute and charming in appearance, are really irritating and annoying pests, and the player never knows when or where he'll see one. And there is no strategy here, no patterns to memorize—all he needs is some good old-fashioned hand-eye coordination. He feels anticipation and suspense as he waits and tries his best to whack those darn moles!

Appendix

A. Cricket Code

```
to mole
loop [
if (random % 8) = 0 [
when [switcha] [
beep
whenoff
]
a, setpower <power>, onfor <time>, brake
whenoff
wait 10
]
end
```

Note: `<power>` and `<time>` are variables which we have to calibrate based on the power in the battery.

B. Other Preliminary Sketches

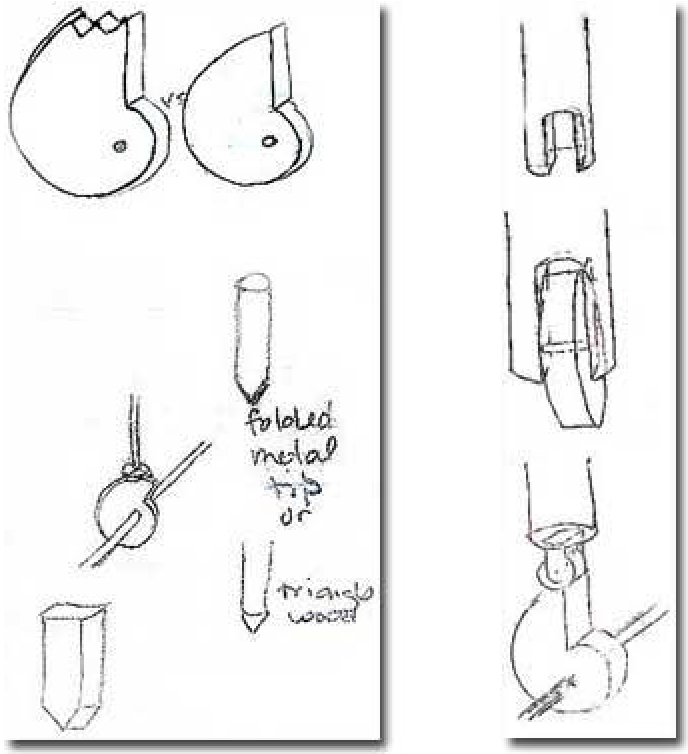

Figure 5: Various Snail Cam Ideas

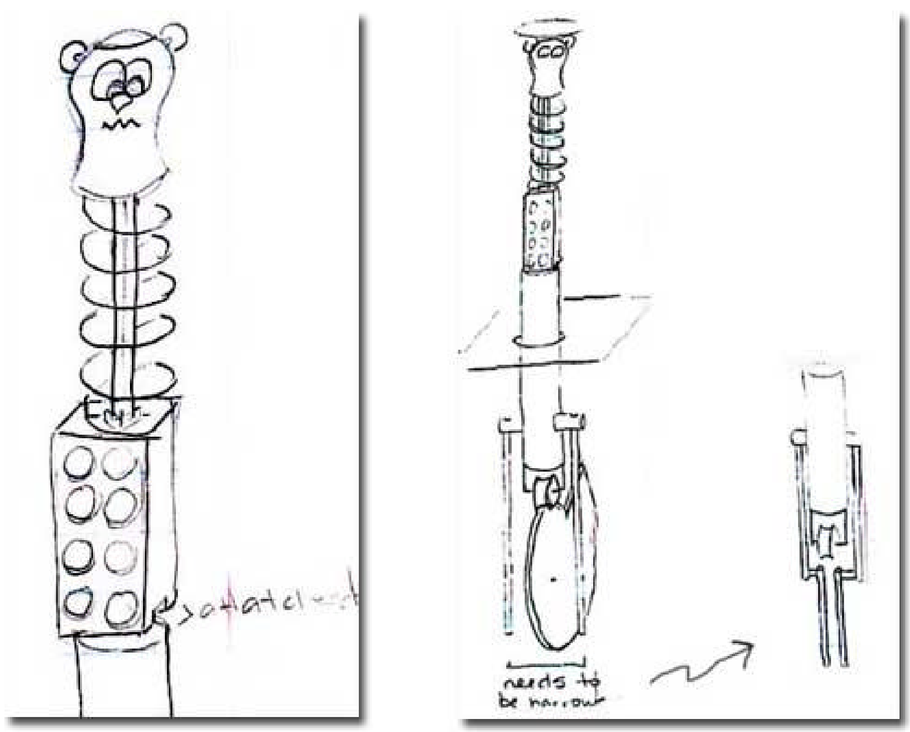

Figure 6: Early Touch Sensor Mechanism and Cam Follower with Guides

C. Other Photos

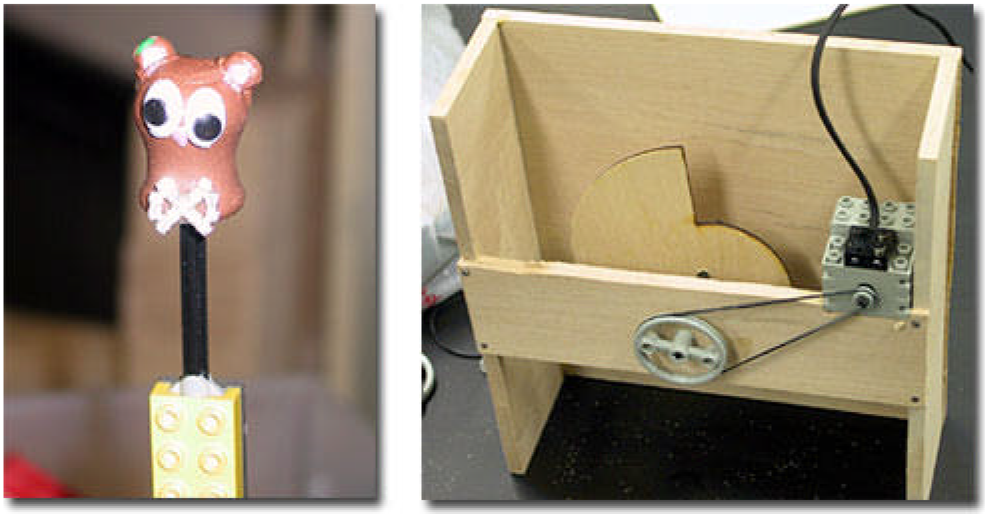

Figure 7: Close-Ups of Mole and Mechanism

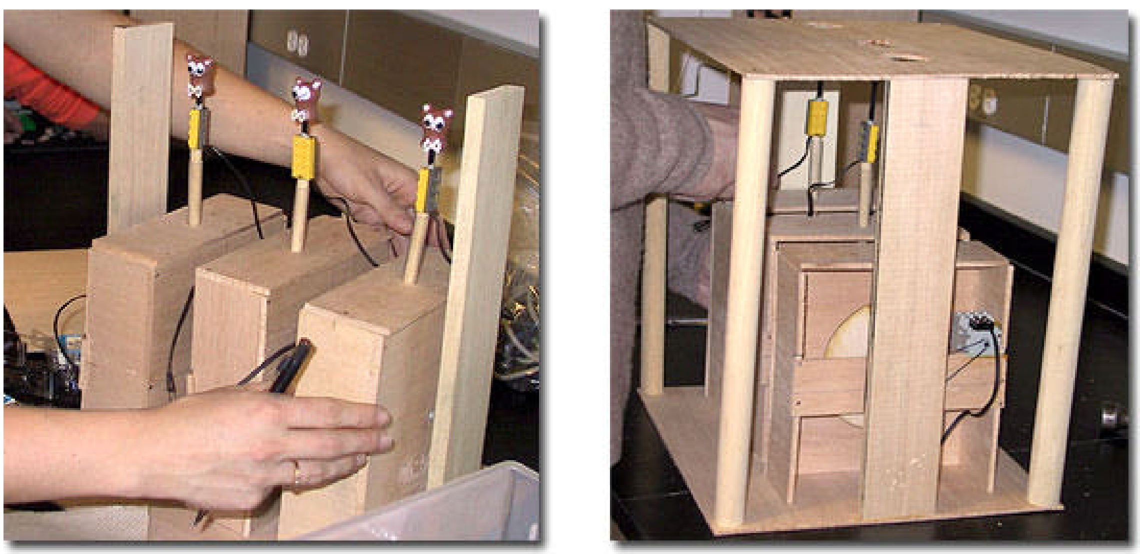

Figure 8: Putting It All Together