MASSACHUSETTS INSTITUTE OF TECHNOLOGY
MEDIA LABORATORY

EPISTEMOLOGY AND LEARNING GROUP

E&L Memo No. 13
June 1991

Braitenberg Creatures

David W. Hogg
Fred Martin
Mitchel Resnick

Abstract

This paper describes 12 autonomous "creatures" built with Electronic Bricks. Electronic Bricks are specially-modified LEGO bricks with simple electronic circuits inside. Although each Electronic Brick is quite simple, the bricks can be combined to form robotic creatures with interesting and complex behaviors, similar to the fictional machines described in Valentino Braitenberg's book *Vehicles* (1984).

Epistemology and Learning Group • MIT Media Laboratory
20 Ames Street • Cambridge, MA 02139 • 617-253-0330

Braitenberg Creatures

David W. Hogg, Fred Martin, Mitchel Resnick

MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139
617-253-7143

June 5, 1991

Abstract

This paper describes 12 autonomous "creatures" built with Electronic Bricks. Electronic Bricks are specially-modified LEGO bricks with simple electronic circuits inside. Although each Electronic Brick is quite simple, the bricks can be combined to form robotic creatures with interesting and complex behaviors, similar to the fictional machines described in Valentino Braitenberg's book *Vehicles* (1984).

Introduction

This paper describes a collection of artificial "creatures" made from LEGO bricks. The creatures were inspired by Valentino Braitenberg's book *Vehicles* (1984).

In *Vehicles*, Braitenberg describes a set of thought experiments in which increasingly complex vehicles are built from simple mechanical and electronic components. Each of these imaginary vehicles in some way mimics intelligent behavior, and each one is given a name that corresponds to the behavior it imitates: "Fear," "Love," "Values," "Logic," etc. Braitenberg uses these thought experiments to explore psychological ideas and the nature of intelligence. Progressing through the book, the reader sees very intricate behaviors emerge from the interaction of simple component parts. In a sense, Braitenberg "constructs" intelligent behavior—a process he calls "synthetic psychology."

Our work follows Braitenberg in spirit. However, instead of Braitenberg's imaginary motors and sensors, we have developed a set of real components known as **Electronic Bricks**—LEGO bricks with electronic circuits inside. These bricks can be connected together to form a wide variety of artificial "creatures," much like Braitenberg's vehicles. Of course, our creatures do not reach the levels of complexity that Braitenberg's do. On the other hand, our creatures exist in the real world: people can actually interact with and play with them. Our creatures do not live only in the world of scientific publications.

We designed our creature-construction kit particularly for children. Children can easily connect the bricks together into new configurations, to create new creatures with new behaviors. In doing so, children can explore, in a playful way, the same deep ideas that Braitenberg describes in his book. In trying to construct particular behaviors, children explore the emergence of complex behaviors from simple components. At the same time, children confront fundamental questions about intentionality. Creatures built from Electronic Bricks fall on the fuzzy boundary between animals and machines, forcing students to come to terms with how machines can be like animals, and vice versa (Resnick and Martin 1990).

This creature-construction activity serves as a prototypical example of the "constructionist" approach to learning (Papert 1980). According to the constructionist paradigm, people are most likely to make deep connections with new ideas when they are involved in constructing meaningful artifacts. In this case, children learn important ideas about living systems not just by observing creatures, but by building them.

Electronic Bricks

Electronic Bricks are LEGO bricks in which we have placed digital electronic circuits with inputs and outputs. They fall into three categories: *action bricks* (such as **motor bricks**), *sensor bricks* (such as **threshold sound sensors**), and *logic bricks* (such as **and bricks**). When connected by a wire, the output of one brick controls the input of another.

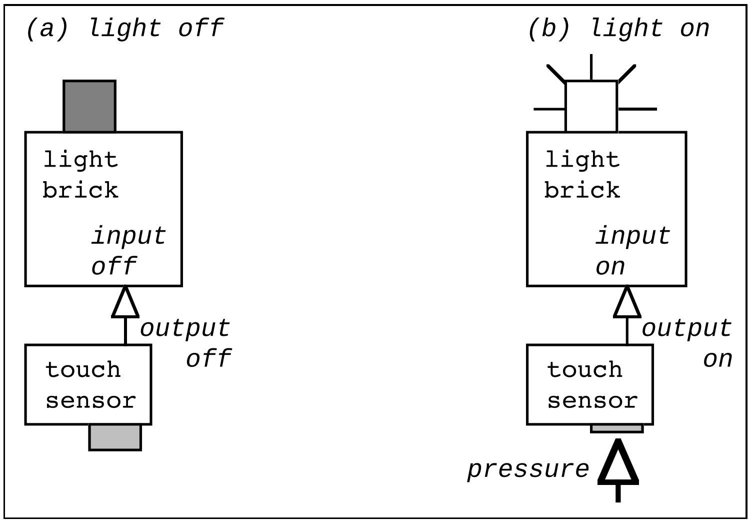

Figure 1: A simple **Electronic Brick** setup.

Imagine that we have a **touch sensor**. Its output is *on* when its button is pressed, and its output is *off* at all other times. If this output is attached to the input of a **light brick**, then the input of the **light brick** will be on when the button is pressed (see figure 1(b)), and off otherwise (see figure 1(a)). This extremely simple setup provides us with a light that turns on when you press the **touch sensor** and turns off again when you let go.

In this paper, we use certain terms that have specific meanings in the **Electronic Brick** universe. These terms will be printed throughout the article in boldface, and they are all defined in the glossary at the end. Included in the glossary are descriptions of each individual type of **Electronic Brick**.

Simple creatures

The first six creatures, *Timid, Indecisive, Paranoid, Dogged, Insecure,* and *Driven*, are all quite simple. They generally have only one sensor and rather limited electronic "brains."

Timid

the shadow seeker

The simplest combination of **Electronic Bricks** possible is one motor attached to one sensor. Happily, even the simplest combinations can produce interesting behaviors.

Timid is a tiny car with one **motor brick** and one **threshold light sensor**, pointing up. The motor brick has two inputs: one controls its power, one controls its direction. The output of the sensor is attached to the *power input* of the **motor brick.** When the light falling on the sensor is in excess of its *threshold*, its output turns on. Because this output is plugged into the power input of the motor, the motor turns on. Hence if sufficient light is falling on the **threshold light sensor**, the car drives forward; otherwise it stands still. The threshold is set with a dial on the side of the sensor.

If the dial on the **threshold light sensor** is set correctly, *Timid* will run when it can "see" the room lights, and stop when it cannot. When the lights are turned on, *Timid* drives until it gets into shadow, at which point it stops. If whatever is casting the shadow is moved, *Timid* will start driving again until it enters another shadow.

Indecisive

the shadow edge finder

Indecisive is identical to *Timid*, except the output of the **threshold light sensor** goes to the *direction input* of the **motor brick** instead of the power input. In this case, the motor is on all of the time, but it goes in one direction when the input is on and in the other when the input is off. This means that *Indecisive* runs forward when it sees the room lights, and backwards when it does not.

When let loose, *Indecisive* drives forward (assuming that the room lights are on) until it gets to a shadow cast by a chair, table, someone's hand, etc. At this point, the **threshold light sensor** no longer sees the overhead lights, and its output switches off. The motor reverses, and the creature runs back into the light. Now the light sensor is illuminated again and *Indecisive* reverts to forward motion, returning to the shadow. In the shadow, the sensor output turns off again, and the vehicle backs up again. It oscillates back and forth at shadow edges.

Paranoid

the shadow-fearing robot

The structure of this creature is what we call a **turtle. Turtles** have two motors, one that drives the wheels on the left side (the *left motor*), and one that drives the wheels on the right side (the *right motor*). If the two motors are doing different things, then the **turtle** will turn. That is, if the left motor is driving forward, and the right motor is off, the **turtle** will move forward and to the right. If the left motor is driving forward, and the right motor is driving backward, the **turtle** will pivot in place to the right. The turtle will move in a straight line only if both motors are on and turning in the same direction.

This particular **turtle** has one sensor, a **threshold light sensor**, whose threshold is set so that its output is on in the light and off in the shadow, just as it is for *Timid* and *Indecisive.* The **threshold light sensor** points up, and it is on an arm that sticks out forward on the vehicle. Its output is connected to the direction input of the left motor. When its output is on, the left and right motors turn in the same direction (the direction of the right motor defaults to forward, the direction of the left motor is set to forward by the output of the **threshold light sensor**), and when the output is off, they turn in opposite directions.

When *Paranoid* is set down in a lit room, it drives straight forward until its protruding **threshold light sensor** enters a shadow. When this happens, the sensor's output switches from on to off and the left wheel reverses. At this point, the left and right wheels are turning in opposite directions. This forces the **turtle** to pivot to the left (see figure 2). It swings around to the left until the protruding sensor has swung back out of the shadow. At this point the left wheel returns to forward motion, and *Paranoid* is off again.

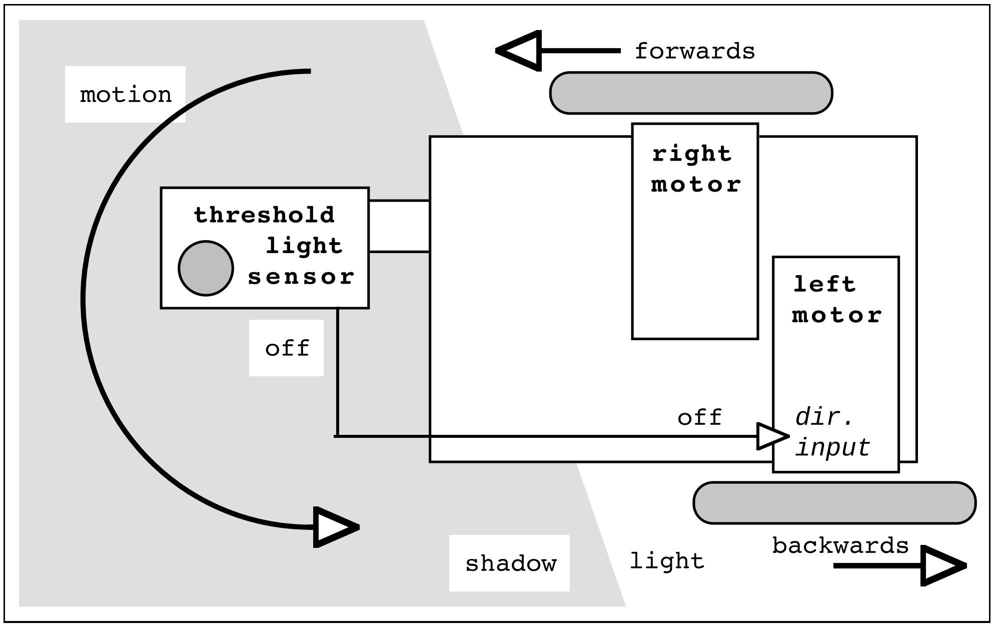

Figure 2: *Paranoid* turning.

Dogged

the obstacle avoider

This vehicle is a small car with a single **motor brick** that drives it forward or backward. It has two **touch sensors**, one facing front and one back, each connected to a bumper. When the bumper is pressed, the touch sensor is activated, and its output turns on. It also has an **or brick** and a **flip-flop brick**. The **flip-flop brick** has two states, an *on state* and an *off state*. In the on state, its ouptut is on continuously, and in the off state, the output is off continuously. It changes state each time its input is turned on. So if the input is off and then turns on, the brick will change state, and it won't change state again until the input turns off and on again. The outputs of the **touch sensors** go to the inputs of the **or brick** and its output goes to the **flip-flop brick**. The **flip-flop brick**'s output is connected to the direction input of the motor.

When *Dogged* is started, it runs either forward or backward, depending on the state of the **flip-flop brick**. When either the front or back bumper is pressed (and hence a **touch sensor**'s output turns on), the **or brick**'s output turns on, which changes the state of the **flip-flop brick**, and the creature reverses direction. In other words, *Dogged* changes direction every time either bumper gets pressed. Let loose, the vehicle will run across the floor until it hits something. When it does, the bumper will be pressed and it will reverse direction and run away. It will continue until it hits something in the other direction. It reverses and runs back to the first obstacle. In this way, it will fall into a pattern of running very quickly back and forth between two objects over and over again.

Insecure

the wall follower

This vehicle is a **turtle**, like *Paranoid*, but it uses the **whisker brick** to sense its surroundings. The **whisker brick** is a brick with a thin plastic strip (*whisker*) pointing out of it. When the strip becomes sufficiently bent, the output of the sensor turns on. This "bending threshold" is set with a dial on the side of the brick. The whisker protrudes over *Insecure*'s left side. The output of the **whisker brick** goes to the power input of the left motor and to an **inverter brick**. An **inverter brick**'s output is on if its input is off, and is off if its input is on. The output of the **inverter brick** is attached to the power input of the right motor. This setup ensures that exactly one motor is on at any time, since the output of the sensor controls one motor and the inverse of that output controls the other.

When *Insecure* is put down in an open area, the **whisker brick**'s output will be off, the right motor will be driving, and it will turn in circles. Imagine, however, that it is put down so that there is a wall close to it on its left. The right motor will drive, turning the vehicle left and forwards, until the whisker touches the wall. Once this happens, the output of the **whisker brick** will turn on, and the left motor will drive; the inverter will turn off and the right motor will stop. *Insecure* will now move forwards and right. Soon it will have turned far enough right that the whisker will no longer be touching the wall, and it will start turning back.

Insecure slowly edges its way along walls and around the bases of pillars.

Driven

the light seeker

Driven is our standard light seeking vehicle. It is a **turtle**. It has a **differential light sensor** mounted on it, facing forward. The **differential light sensor** is a brick with two outputs. When the left side of the sensor is receiving more light than the right side, the *left output* will be on. When the right side is receiving more light than the left side, the *right output* will be on. The left output of the **differential light sensor** is attached to the power input of the right motor, and the right output to the power input of the left motor. Thus, the left motor turns on when the right side of the sensor is brighter, and the right motor turns on when the left side is brighter.

Driven moves towards a bright light by successive right and left turns. If the light is on the left, the left output of the **differential light sensor** will be on and so the right motor will be running, moving the creature forward and rotating it left. Once the vehicle has turned far enough that the light is on its right, it starts moving forward and right. *Driven* slowly wiggles its way towards light sources.

More complex creatures

In this section we present 4 creatures, all of which are a bit more sophisticated than the previous ones.

Persistent

the light seeker with a collision algorithm

This vehicle is the same as *Driven*, but with an attempt to give it a method for avoiding obstacles.

In addition to *Driven*'s **differential light sensor**, *Persistent* has a bumper on its front which is attached to a **touch sensor**. This sensor plugs into a **timer brick**. The **timer brick** has one input and one output.

Each time the input turns on, the output turns on for a period of time and then turns off again. The period is set with a dial on the side of the brick. The output of the **timer brick** is attached to the direction inputs on the motors.

When the creature is let loose, it starts moving towards the light source in the same way that *Driven* does. However, if it collides with something on its way, the bumper will get pressed. When this happens, the **touch sensor** will be activated, the **timer brick** will turn on and the motor directions will be reversed. They will stay reversed for the duration of the **timer brick**'s period.

Upon collision, *Persistent* backs up for a short period of time and then the **timer brick** switches off. The creature resumes travelling towards the light. However, *Persistent* fails in its goal of avoiding obstacles, since the creature generally resumes travelling along the same path it was on when it made the contact. It usually collides with the same object that it did before, and it does this over and over again.

Attractive and Repulsive

The leading and following pair

This example actually involves two Braitenberg creatures.

Attractive is a creature with one **motor brick** and a **threshold light sensor** pointing towards the rear. The **threshold light sensor**'s output goes to the power input of the motor. It drives forward when the sensor's output is on. *Attractive* is a very quick vehicle.

Repulsive is a slow creature with one **motor brick** and a bank of bright lights. It continually drives forward. When *Repulsive*'s lights get sufficiently close to *Attractive*'s **threshold light sensor**, the output of the sensor will turn on, and *Attractive* will move.

Imagine that *Attractive* and *Repulsive* are set down in a line, with *Repulsive* facing *Attractive*'s back. *Repulsive* will drive slowly forward until its lights come within the range of *Attractive*'s sensor. The sensor's output will turn on and *Attractive* will run quickly away from *Repulsive* until it is out of range and so the motor stops again. Soon, however, *Repulsive* will have come within range again.

Consistent

the four-state turtle

This vehicle is a **turtle** with one **threshold sound sensor.** When the **threshold sound sensor**'s tiny microphone "hears" a sufficiently loud noise, its output turns on for a very short time. Its threshold is set with a dial on the side of the brick. The sensor's output goes to a **flip-flop brick.** The output of that **flip-flop brick** goes to another **flip-flop brick**, as well as the direction input of the left motor. The second **flip-flop brick**'s output goes to the direction input of the right motor.

Every time the **threshold sound sensor** is triggered, the output of the first **flip-flop brick** changes state (either from on to off or off to on). So, every *second* time the sensor is triggered, the output of the first **flip-flop brick** turns from off to on (as opposed to changing from on to off). Whenever the output of the first **flip-flop brick** changes from off to on, the output of the second **flip-flop brick** changes state. In other words, the output of the first **flip-flop brick** changes state every time that the sensor is triggered, and the output of the second **flip-flop brick** changes every second time. This means that *Consistent* has four different "mental" states: on-on, off-on, on-off, off-off.

Because the output of the first **flip-flop brick** is attached to the direction input of the right motor, and the output of the second is attached to the direction input of the left motor, these "mental" states each correspond to one of four states of motion. The first corresponds to forward motion, the second to turning left, the third to turning right, the fourth to moving backwards. If one claps loud enough in the vicinity of

Consistent, it will change state. In fact, as one claps repeatedly, *Consistent* will cycle through its four states in this order.

Inhumane

the mousetrap

Inhumane is designed to capture mice. It consists of a long, thin tunnel with a door at one end. The door can be moved by turning on a **motor brick.** Near the other end of the tunnel, there is a light source on one side and a **threshold light sensor** on the other. Some mouse food is placed at the very end, past the light beam and sensor. If a mouse enters the chamber, he or she must break the light beam to get to the bait. The light is always on, and the sensor's threshold is set so that the output is on when the light's path is unobstructed, and off when it is blocked.

The layout of **Electronic Bricks** that make up *Inhumane*'s "brain" is shown in figure 3.

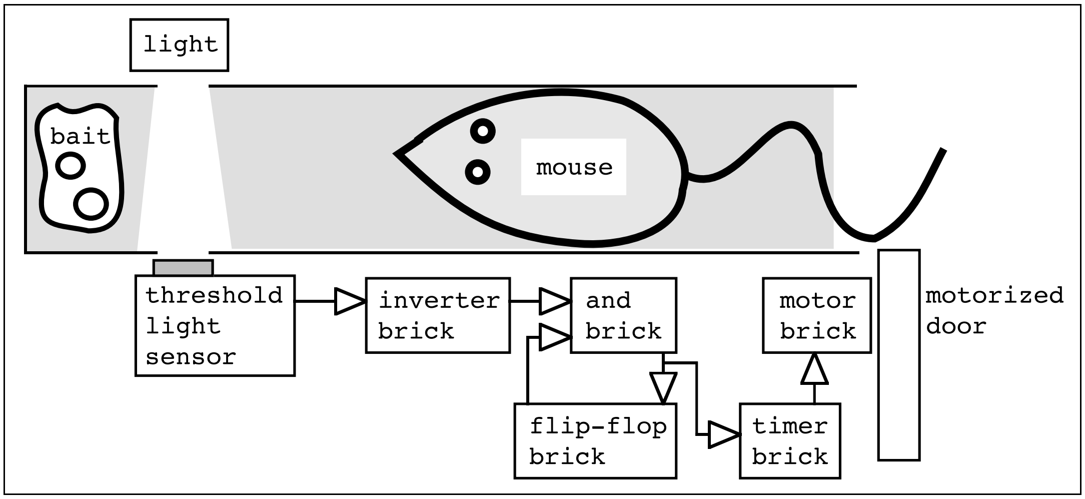

Figure 3: The layout of *Inhumane*.

Imagine that we start with the trap's door open and the **flip-flop brick**'s output on. In this situation, one of the **and brick**'s inputs is on. If the beam is broken by a mouse, the **threshold light sensor**'s output changes to off, the **inverter brick**'s to on, and then both of the **and brick** inputs are on. The **and brick** turns on the inputs of the **flip-flop brick** and the **timer brick.** The **timer brick**'s output turns on for just long enough to close the door. The **flip-flop brick** changes state and therefore one of the inputs to the **and brick** is now off.

Now the mouse is incarcerated, and the circuit is impotent. The mouse can run back and forth past the beam all he or she likes, but with one input to the **and brick** off (the one from the **flip-flop brick**), the **timer brick** will never turn on again. In other words, when the **flip-flop brick** is in its off state, the **threshold light sensor** has no effect on it, whereas when it was in its on state, the sensor could indeed affect it. This circuit is directly analogous to a physical trap, because most traps begin in a loaded state (**flip-flop brick** on), get triggered (light beam is broken), and then close, entrapping the victim (**flip-flop brick** off). After the trap is sprung, no amount of fiddling with the trigger will reopen it (re-breaking the beam does not change the **flip-flop brick**'s state).

In practice, *inhumane* caught several mice in our building. Most were able to chew their way out, leaving a pile of LEGO dust behind. However, we did indeed relocate one mouse to a more natural habitat. Soon, however, our creation was outmoded by more efficient methods of pest control.

Philosophical creatures

These two machines value thought over action. The first, *Frantic*, does nothing but think: it only observes itself. The second, *Observant*, observes its environment, but does not change it.

Frantic

the negative feedback loop

Frantic consists of a **light brick**, an **inverter brick** and a **threshold light sensor**. A **light brick** is an **Electronic Brick** with one input and a small light. When the input is on, the light is on; when the input is off, the light is off.

The **threshold light sensor** points at the **light brick**, and its output goes to the **inverter brick**. The **inverter brick**'s output goes to the **light brick**. Now there is a "paradox": if the light is on, the sensor's output is on, the **inverter brick**'s output is off, and the light turns off. If the light is off, then the sensor's output turns off, the **inverter brick**'s output turns on, and the light turns on.

As one might predict, the light blinks on and off at a frenzied rate.

Observant

the creature sensitive to the direction of a sound

By comparing the outputs of two **threshold sound sensors**, *Observant* can find the direction from which a sound comes. The circuit does not lend itself to verbal description, so we present a diagram: figure 4.

In air, sound travels a foot in about one millisecond. With the **threshold sound sensors** approximately one foot apart, *Observant* can indeed tell the direction from which the sound came since the sound takes about a millisecond to get to the more distant sensor. That is, if the sound comes from the left, and hits the left sensor first, the output of the left **timer brick** will turn on, and if it comes from the right, the output of the right **timer brick** will turn on. The circuit restores itself when the timing cycles of the **timer bricks** run out.

It is interesting to note the conceptual similarity between the electronics in *Inhumane* and *Observant.* Both creatures have circuits that start in one state, and in the presence of some stimulus "drop" into some new state which the initial triggering stimulus is unable to change. Both circuits are analogous to traps. But because *Observant* uses **timer bricks** instead of **flip-flop bricks**, it restores automatically to its initial state after a specific time.

Acknowledgments

The creatures described in this paper were designed by Ryan Evans, David Hogg, Fred Martin, Seymour Papert, Mitch Resnick and Brian Silverman.

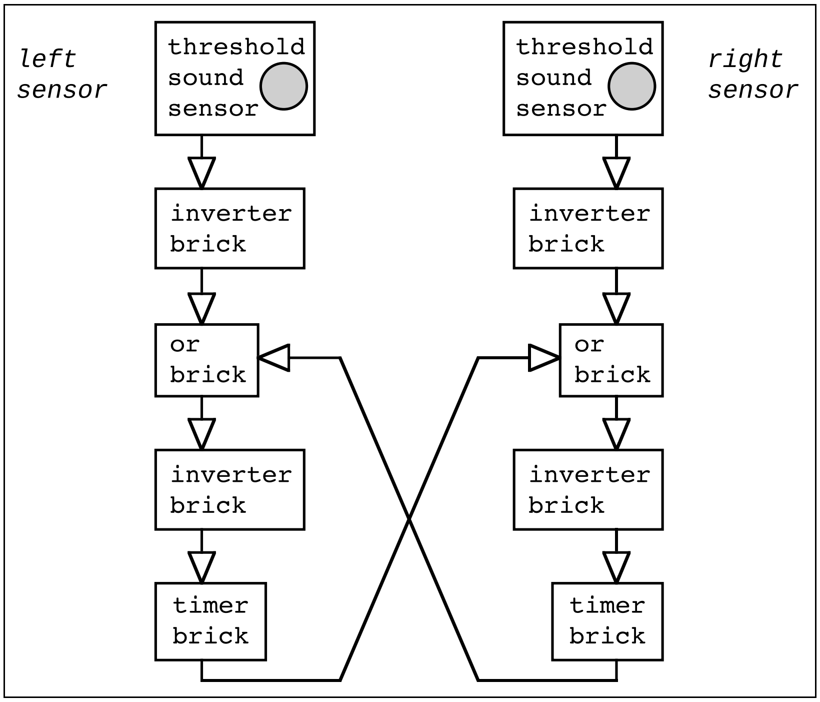

Figure 4: *Observant's* "brain."

References

[1] Braitenberg, Valentino (1984) *Vehicles: Experiments in Synthetic Psychology.* The MIT Press. Cambridge, Massachusetts.

[2] Papert, Seymour (1980) *Mindstorms: Children, Computers, and Powerful Ideas* Basic Books, New York.

[3] Resnick, Mitchel, and Martin, Fred (1990) "Children and Artificial Life," E&L Memo No. 10, MIT Media Laboratory. Cambridge, Massachusetts.

Glossary

And brick: An **Electronic Brick** with two inputs and one output. If both inputs are on, the output is on; otherwise it is off.

Electronic Bricks: These are the subject of this paper. They are LEGO bricks with simple electronic circuits inside, and *inputs* and *outputs* that allow you to connect them together. The *inputs* and *outputs* are digital (only on or off). There are *action bricks* (such as **motor bricks**), *sensor bricks* (such as **threshold sound sensors**), and *logic bricks* (such as **and bricks**). When connected by a wire, the *output* of one brick controls the *input* of another.

Differential light sensor: An **Electronic Brick** with two light sensors and two outputs. When the sensor on the left side of the brick is more brightly lit than the right, then its *left output* is on, if the right is brighter, then its *right output* is on. If the light levels on the two sides are extremely close, then both outputs are on.

Flip-flop brick: An **Electronic Brick** with one input and one output. This brick has two states: an *on state* and an *off state*. When the input to this brick turns on, it changes state. That is, if it is in the *off state* and the input changes from off to on, it changes to the *on state*, and it will stay that way until the input turns off and then on again. When this happens, it will change back to the *off state*. When in the *off state*, its output is off, and in the *on state*, its output is on. The **flip-flop brick** is like a toggle switch.

Motor brick: An **Electronic Brick** with two inputs, and a motor shaft which can be attached to wheels or other LEGO machinery. If the *direction input* is on, the shaft turns in one direction. If it is off, it turns in the other. If the *power input* is on, it runs. If it is off, it does not. *Note:* If there is nothing attached to the power input, the motor defaults to on.

Inverter brick: An **Electronic Brick** with one input and one output. When the input is on, the output is off. When the input is off, the output is on.

Light brick: An **Electronic Brick** with a tiny lightbulb. It has one input. When the input is on, the light is on; when the input is off, the light is off.

Or brick: An **Electronic Brick** with two inputs and one output. If both inputs are off, the output is off; otherwise it is on.

Threshold light sensor: An **Electronic Brick** with a light sensor and one output. When the light falling on the sensor is above a certain level, the output is on; otherwise it is off. The threshold is set with a dial on the side of the brick.

Threshold sound sensor: An **Electronic Brick** with a tiny microphone and one output. When the sound picked up by the microphone is sufficiently loud, the output turns on for a short time. The threshold is set with a dial on the side of the brick.

Timer brick: An **Electronic Brick** with one input and one output. Each time the input turns on, the output turns on for a period of time. The period is set with a dial on the side of the brick.

Touch sensor: An **Electronic Brick** with a button on its side and one output. When the button is depressed, the output is on; otherwise it is off.

Turtle: A type of vehicle with wheels on the left side and wheels on the right side. **Turtles** have two **motor bricks**, one that drives the wheels on the left side (the *left motor*), and one that drives the wheels on the right side (the *right motor*). If the two motors are doing different things, then the **turtle** will turn. For example, if the left motor is driving forward, and the right motor is off, the **turtle** will move forward and to the right. If the left motor is driving forward, and the right motor is driving backward, the **turtle** will pivot to the right in place. It will only move in a straight line if both motors are on and turning in the same direction.

Whisker brick: An **Electronic Brick** with one output and a thin plastic strip sticking out one side. When the strip gets bent far enough, the output turns on; otherwise it is off. The bending threshold is set with a dial on the side of the brick.