## **Rube Goldberg Machine**



# **Design Project**

Inwoo Kim Jane Lee Ben Moseid

Water Calculator

Engr100A (Prof. Wei-Chih Wang)

## You want to do what?!

### **Objective:**

To design a simple machine that engages in a complex chain reaction to accomplish a simple task...

### **Given Criteria:**

- There must be at least 20 steps in the process
- The machine must accomplish its purpose
- It can only occupy an area 1.5 m x 1.5 m on the floor

Water

Calculato

- It cannot take too long to accomplish its ends
- It must be complete in 3 weeks time

### **Personal Goals:**

- Be interesting and complex
- Be new, seldom or never done before

Water

Calculator

- Exploit and compliment each group member
- Use the cool Lego set
- Get a good grade

## Kicked to the curb...

Water

Calculator 🗑

- **Alternate ideas:**
- trap > overdone
- letter head writer > too complicated
- fish feeder > dumb
- light switch > too simple

## What we came up with...

A water calculator that will be able to add an 'x' amount of bits together through a series of binary representations that are processed through a device that channels water.

Water

Calculator



## ...and this is why...

- The idea fit the given criteria and personal goals
- The theoretical concept often differs from the realistic implementation, therefore a prototype design was needed

Water

alculator

## **Original Prototype Design**



The colored water would flow down the various plastic tubing's.

It would either end up in the final result or the carry over.

The process is repeated for the number of bits that need to be calculated.

Water Calculator

## The Number System

Water

alculator

- Decimal Numbers
- 6357 has 4 digits.
- 7 is filling the 1's place
- 5 is filling the 10's place
- 3 is filling the 100's place
- 6 is filling the 1000's place

- So mathematically it could be expressed:
- (6\*1000)+(3\*100)+(5\*10)+(7\*1) = 6357
- Or it can be expressed in the powers of 10
- (6\*10^3) + (3\*10^2) + (5\*10^1) + (7\*10^0)
  =6357

Water

alculator

• Since we have 10 fingers we naturally used the base 10 system.



- Computers however, tend to use binary digits. It is easily represented by either, current going through the wire, or not going through the wire.
- So just like it is in Decimal;
- The number 1011 in binary would be:

Water

alculator

- (1\*2^3)+(0\*2^2)+(1\*2^1)+(1\*2^0) =
- 8 + 0 + 2 + 1 = 11

## **Boolean Logic**

- Now that we know how Binary numbers work, there are various rules to simple "switches" (gates)
- Simplest of all gates is called a NOT gate. Whatever is inputted 1/0, it will output the exact opposite.

Water

alculator

- The gate we used in this adder is called the AND gate.
- The AND gate effectively takes 2 values and gives a Boolean result with these values.
  - A | B | Q
    0 | 0 | 0
    0 | 1 | 0
    1 | 0 | 0
    1 | 1 | 1

Water 4

Calculator

- Another gate we used is called an OR gate
- Just like the AND gate, it takes 2 values and returns 1 value

A | B | Q
0 | 0 | 0
0 | 1 | 1
1 | 0 | 1
1 | 1 | 1

Water

Calculator 🛛

## **Binary Math**

- Binary math is just like decimal math, but lets start with decimal addition
- 452
- <u>+751</u>
- 1203

First its 2+1=3; then its 5+5= 10, so we carry over the 1. 1+4+7= 12, so we carry over again, 0+0+1 = 1, so that is the result

Water Calculator

- Binary math is exactly the same except for we only use 0's and 1's.
- 010 2
- <u>+111</u> +7
- 1001 9
- First 0+1=1; then 1+1=10, so there is a carryover of 1. 1+1 = 0, another carryover, 0+1 = 1.
- If we convert that back to Decimal it would be (1\*2^3)+(1\*2^0) = 9

Water •

Calculator

## **Our Water Calculator Solution**

 We decided to use a simple AND OR combination (x2) to create a simple 1 bit adder + a carryover bit. We could serialize the components to do an unlimited amount of bits, just by reading the carryover bit and repeating the process.

Water

alculato



# **Actual Design**

#### **LEVEL1- Water Reservoir:**



- A 2-liter bottle with a hole cut out towards the "top"
- 1/4" tube attached to the neck of the bottle



Water Calculator

### **LEVEL2-** Water Dispenser:





- <sup>1</sup>/<sub>4</sub> turn valve
- Lego Mindstorm software
- Motor mounted to a plane so that shaft of motor is perpendicular to platform
- Gears, large and small

Water Calculator

 Intake tube and directional tube

10100010101001010101010101

00100101000<sup>.</sup>

### Water Dispenser cont...



 Proponent mounted on a "cart"

 Dispenser moves on 4" x 18" Lego track

 All materials adhered by hot glue gun

Water Calculator

### **LEVEL3- Belt:**





- 10" x 32" rubber belt
- Lego skeleton powered by 2 Mindstorm motors
- 3" wheels
- Gears- 3 large, 3 small
- Line of 3 cups mounted onto belt with popsicle sticks, hot glue gun, wiring

00010101001010101010101

Water Calculator

### Belt cont...



Belt apparatus
mounted on top of
wood box-like frame 4"
x 12" x 14" with 2 14"
wooden sticks



#### LEVEL4- Water "Switch" 1:



- Two 4" funnels, one 3" funnel mounted with popsicle sticks
- Two 1/2" tubes cut to 6" in length
- 7" x 10" wood board to set tubes at an 135% angle
- One 6" plastic bowl with 2" of bottom cut off sitting on top of 6" funnel



#### Water "Switch" 1 cont...





- One defect overflow funnel
- One 8oz cup with 1.5" holes on both sides and at bottom for water
- Two 3/8" tubing at 30" length
- Two RCX Bricks
- Rotary sensor for input mechanism
- All materials adhered with hot glue gun

Water Calculator

#### LEVEL5- Water "Switch" 2:



- One 6" plastic bowl with bottom cut off
- One 6" funnel, one 3" funnel mounted with popsicle sticks
- 3" x 6" wood board holding "switch" 1 "true" tube and overflow tube in place
- All material adhered with hot glue gun

Water Calculator

### **LEVEL6-** Results/ Carry Over:



- 1 18" x 4 "x ½" Particle Board
- Eight, 8oz cups
- Lego, 4 large wheels
- RWD motor/ Mindstorm
- Mindstorm light sensor mounted in a Lego tower
- Two 1" pipes 6" in length encapsulating "true" and "false" tubes held by popsicle sticks

Water Calculator

# Prototyping

 One of the most important prototype performance failures we ran into was the timing portion of the software. It was critical that every timing element was correct down to the 1/100th of a second. This was one of the expected prototype failures that we would need to iron out. Otherwise, the only other failure was the unexpected fluid dynamics of the water streams when they collided with each other.

Water

Calculato

## Conclusion

- In starting the project, we made a conscious decision to incorporate a simple, very sequential adder. This was achieved by having the least amount of physical "switches"; however, after doing this design, we think it would be better to make a wide, parallel design.
- It would cut down on the timing aspect of the project, thus greatly increasing the reliability of the system.

Water

alculato

## **Conclusion continued**

- On the downside, it would require at least 8x more switches, at the expense of that reliability.
- Overall, it was a good learning experience of how water could theoretically replace electricity for Boolean logic. Since the timings were greatly reduced for transmission, it was very difficult to use the resources we had, to implement a way to get it to correctly function.

Water

alculato