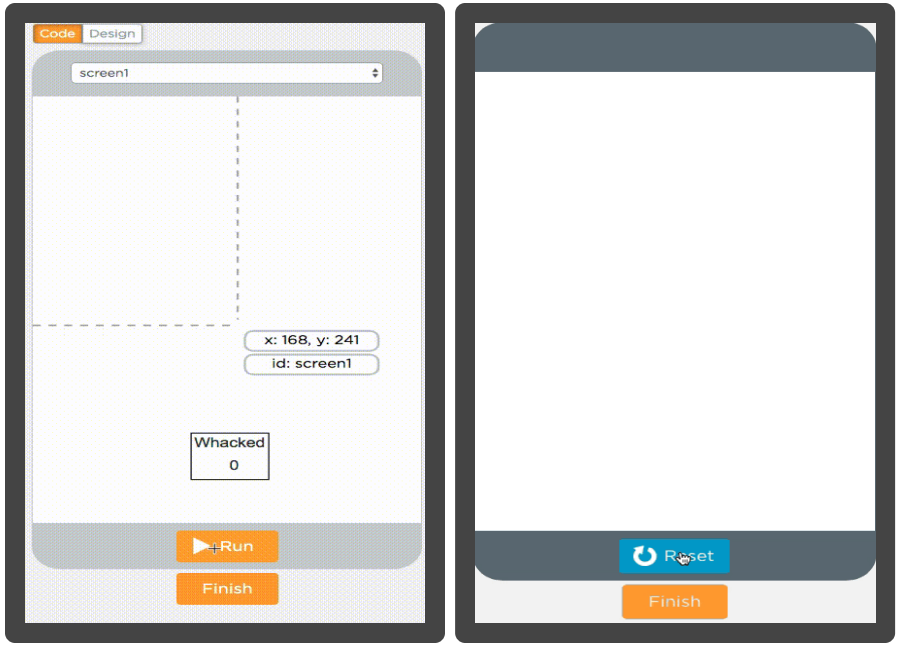
Programming and fun

Why program?

Solve a lot of interesting problems

- Building games, legos, mars rovers.
- Build control software for things like automated cars
- Solving tough mathematical problems like “safe prime numbers”.


Code
Design
screen1
x: 168, y: 241
id: screen1
Whacked
0
Run
Finish
Reset
Finish

Why program?

For systems that power AI and Machine Learning.

Sometimes for more controversial purposes

- Behavior modification: Like getting you to buy things online or influencing decisions
- With great power comes great responsibility!

Earn a multi 6 figure salary with a few years of work.

What am I going to tell you today

Introduce to you some project ideas using a few games and (if time permits a bit of math.)

Get you generally interested in programming

Game 1: Whack a mole.

First gather “requirements”

```
- You want X number of moles to pop out randomly.
- You want to be able to "hit" them with a mouse click.
- You want to know at the end how many you moles you hit.
```

Game 1: Whack a mole(First attempt).

There are a few ways to do it

You can draw a mole's picture using functions like drawImage(id, x, y, w, h) which is a canvas function:

```
1 createCanvas();
2 drawImage("mole", 20, 20, 60, 40);// or us drawImageURL(...);
3 hideElement("mole");
```

This approach has a drawback.

Game 1: Whack a mole.

Better Approach: use a different API.

You can use `setPosition()` to set the mole in `X,Y` plane on the screen.

```
1 function startGame(time) {
2   timedLoop(time, function() {
3     // Get a random position on X, Y
4     // Show the mole using setPosition
5     setPosition("mole", x, y, h, w);
6   }
7 }
```

Bonus: Can you add additional code that counts the number of whacked moles? - This is the part where using `setPosition` really shines.

Game 1: Whack a mole(lessons learnt).

What API do I use?

So when should you use canvas based functions?

How to use basic javascript functions like `timedLoops` and `onEvent` to build full games.

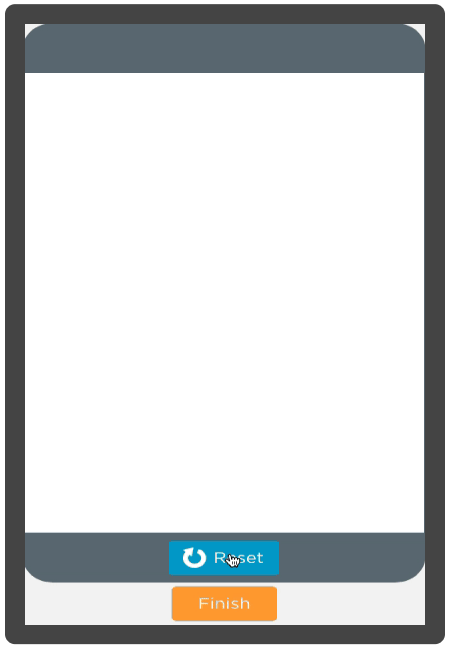
Game 2: Minesweeper.

First gather “requirements”(i.e. what the game's core functions should do)

- A M*N minefield with certain % having a mine.
- clicking a square reveals the number of mines around it.
- Not all squares have mines around it.
- Place flags where you think there is a mine.

We won't be able to implement all of the features, but get pretty close to the real thing.

Playing Minesweeper.



Minesweeper: Baby steps

How does the board stored in memory?

An Array of Arrays or a matrix

```
1 var board = [
2   ['M', 'M', 'E', 'E', 'E'],
3   ['M', 'E', 'M', 'E', 'E'],
4   ['E', 'E', 'M', 'E', 'E'],
5   ['M', 'M', 'E', 'E', 'M'],
6   ['M', 'E', 'E', 'E', 'M']]
```

Minesweeper: Baby steps

How does one generate an empty board?

```
1 function generateEmptyBoard(rows, cols) {
2   board = []
3   for(var row = 0; row < rows; row++) {
4     board.push([])
5     for(var col = 0; col < cols; col++) {
6         board[row].push("")
7     }
8   }
9 }
```

Using the basic structure of loop with in a loop can u print a board to the console?

Can you guess how you can place mines on the field?

Minesweeper: Graphics is not so hard.

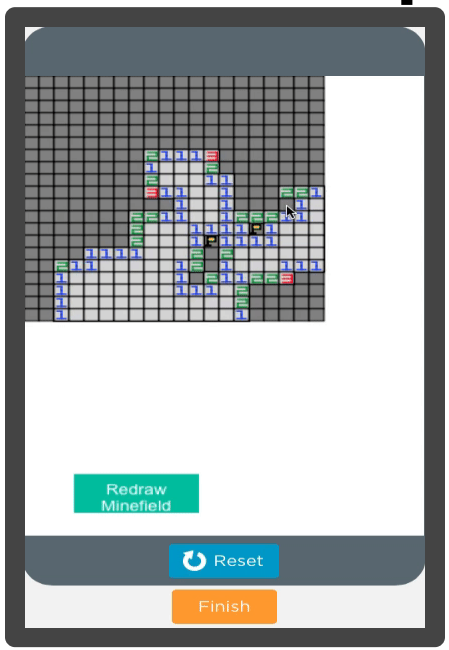
Drawing a field:

```
 1 NROW = 10;
 2 NCOL = 10;
 3 S = 6;
 4 setFillColor("gray")
 5 setStrokeColor("black");
 6 for(var i = 0; i < NROW; i++) {
 7   for( var j = 0; j< NCOL; j++){
 8     // Calculate the positions on the field on paper to get the formula
 9     // i    j   x   y
10     //  0   0   0   0
11     //  1   0   0   8
12     //  2   0   0   16
13     //...
14     //  1   0   8   0
15     //...
16     var x = S*j;
17     var y = S*i;
18     rect(x,y, S, S);
19   }
20 }
```

Drawing mines are also easy take the center and draw a black circle.

Drawing numbers and flags requires a bit more geometry.

Minesweeper: Blowing a mine.



Minesweeper: Graphics is not so hard.

Key: You need to translate row and col integers to pixels

```
1 function rowColToCanvasCoordinates(row, col) {
2     return [S*col, S*row,
3     S+S*col, S+S*row];
4 }
```

And vice versa:

```
1 function eventCoordinatesToRowCol(x, y) {
2     var row = parseInt(y/S);
3     var col = parseInt(x/S);
4     // Null if the click is OOB
5     if(row >= NROW || col >= NCOL) {
6         return null
7     }
8     return [row, col];
9 }
```

Challenge: Can you use `rowColToCanvasCoordinates` to draw the empty and mined cells on the canvas?

Minesweeper: Wrap up

You learn't how to represent a game board in memory.

How to traverse it.

How to translate the game board and its cells into graphics.

Now you can build other games like Battleship, tic tac toe using these basic pieces.

Thank you!