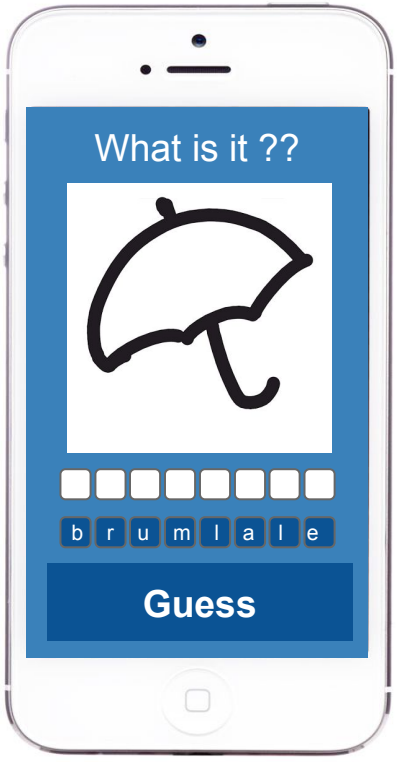
Pictionary App: Communicating Pixels

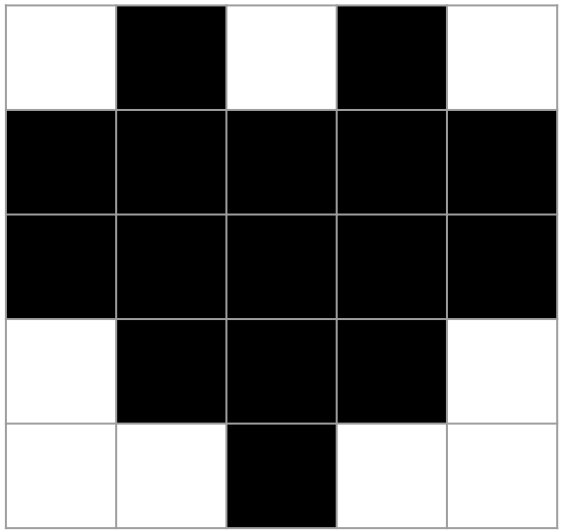
Pictionary apps!!


```
10000111011101000011011 0
11010100000101111100000 0
01001000010111110000000 0
01111000001001111010000 0
00001000111111101110000 0
00000101110100101000010 1
```

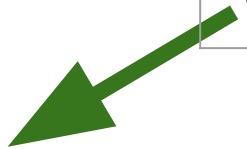
Data transferred by the magic of the internet as ones and zeros!



Images represented as 1’s and 0’s



0	1	0	1	0
1	1	1	1	1
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0



```
0101011111111110111000100
```

Recreate a drawing app without phones!

Find another team.

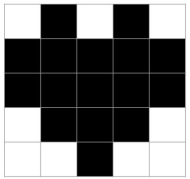
Send each other messages!!

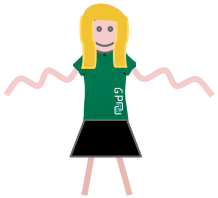
TEAM 1

1. Create drawings
2. Turn them into 1's and 0's.

Draw a heart

```
0101011111111
110111000100
```





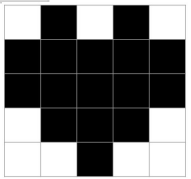
Transfer to the other team using the magic of ~~internet~~ tutors!

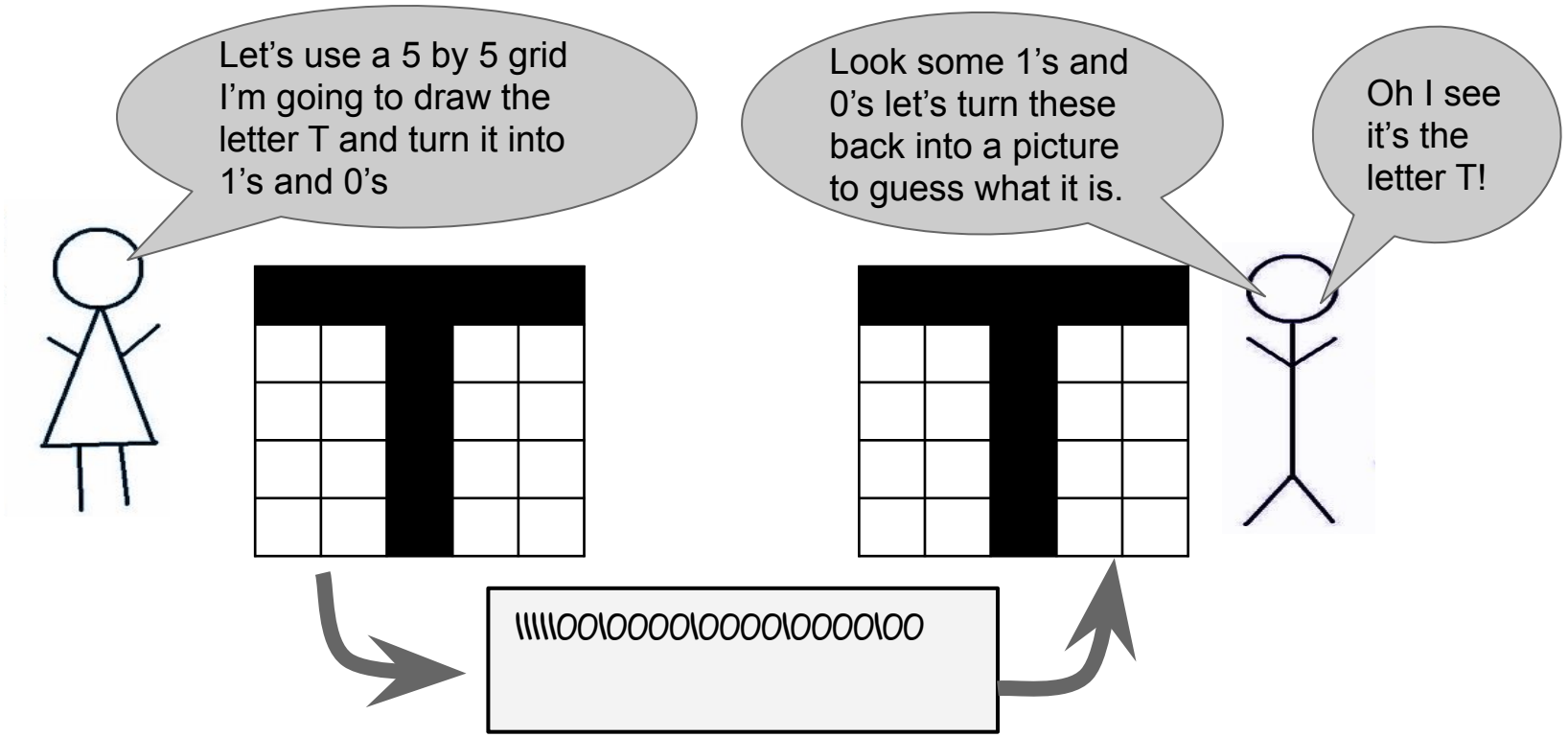
TEAM 2

1. Transform them back to pictures
2. Guess what they are!

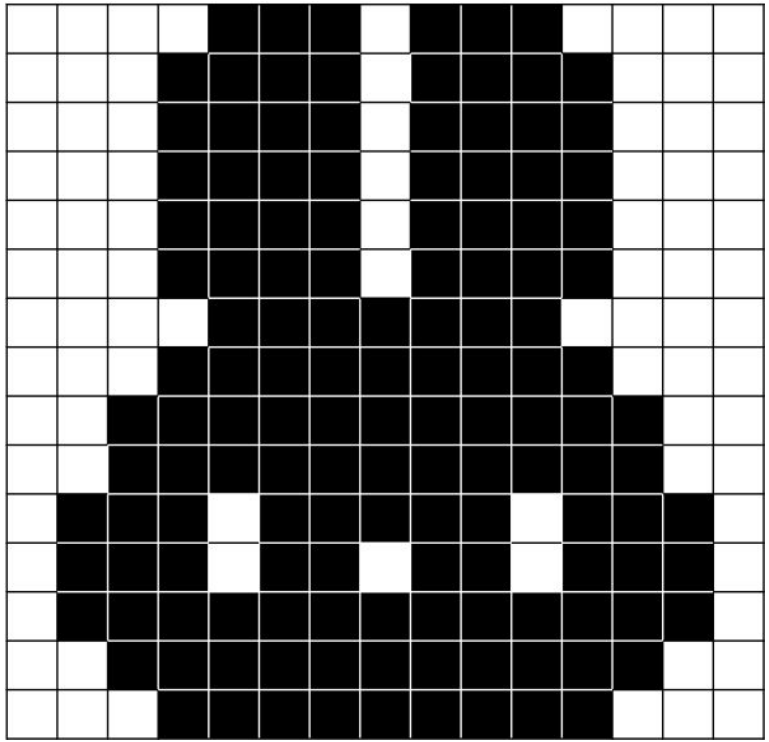
```
0101011111111
110111000100
```

It's a heart!!!




Let's use a 5 by 5 grid I'm going to draw the letter T and turn it into 1's and 0's
Look some 1's and 0's let's turn these back into a picture to guess what it is.
Oh I see it's the letter T!
\\\\\OO\OOOO\OOOO\OOOO\OO

But what if we need more squares?



The string for this bunny is 225 characters long:

```
000011101110000000111101111000000111101
111000000111101111000000111101111000000
111101111000000011111110000000111111111
000001111111111100011111111111000111011
111011100111011011011100111111111111110
001111111111100000111111111000
```

That's too many to deal with. We're going to make a mistake!

We can compress our messages!

The bottom row of the bunny looks like this:

000111111111000

3 0s 9 1s 3 0s

This can be written as **0**3**1**9**0**3

We can compress our messages!

The second bottom row of the bunny looks like this:

0011111111111100

2 0s 11 1s 2 0s

This can be written as **0**2**1**11**0**2

We can compress our messages!

The second bottom row of the bunny looks like this:

0011111111111100

2 0s 11 1s 2

This is a problem! Is it 1 lot of 1 or 11 lots of 1? Our system breaks!

This can be written as **0**2**1**11**0**2

But we have a problem!

Solution! (Well partly)

The second bottom row of the bunny looks like this:

0011111111111100

2 0s 11 1s 2 0s

What if we used letters?

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Solution! (Well partly)

The second bottom row of the bunny looks like this:

00111111111100

2 0s B 1s 2 0s

NOW this can be written as **0**2**1**B**0**2

Solution! (Even Better!)

The second bottom row of the bunny looks like this:

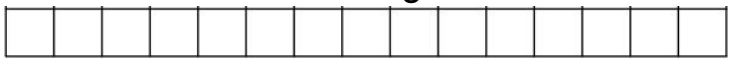
00111111111100

2 B 2

We don’t need to say 0 or 1. We can just write 2B2. *We’ll just agree to always start drawing from blank (or 0)*

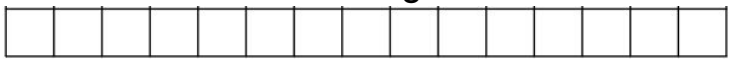
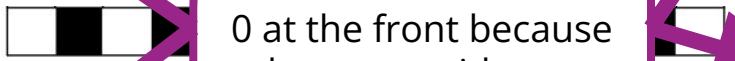
Examples

Representing different things in Run Length Encoding

A whole white line of length 15 	We always start by representing blank so this would just be F (meaning 15 blanks)
A whole black line of length 15 	We always start by representing blank so this would be 0F (zero blanks and 15 blacks)
A checkerboard starting with white 	111111111111111
A checkerboard starting with black 	0111111111111111 - this is the most annoying thing to represent but we're not going to use it if we can help it

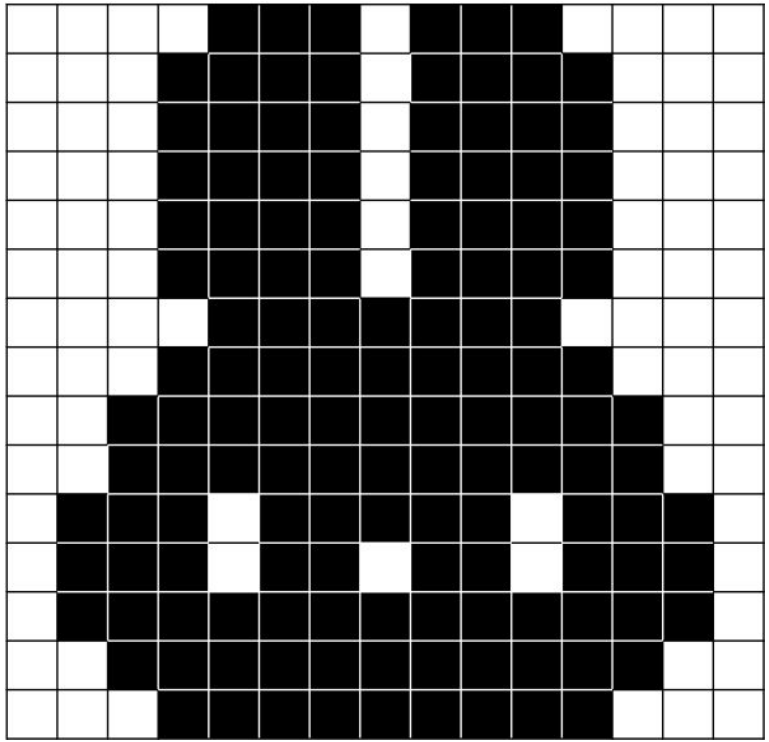
Examples

Representing different things in Run Length Encoding

A whole white line of length 15 	We always start by representing blank so this would just be F (meaning 15 blanks)
A whole black line of length 15 	We always start by representing blank so this would be 0F (zero blanks and 15 blacks)
A checkerboard 	111111111111111
A checkerboard 	0111111111111111 - this is the most annoying thing to represent but we're not going to use it if we can help it

Notice that rows that start with black need a 0 at the front because they start with zero blanks

Run length encoding the bunny



43134 - 34143 - 34143 - 34143 - 34143 - 34143 - 474 - 393 - **2B2** - **2B2** - 131513 - 13121213 - **1D1** - **2B2** - 393

Isn't that easier than

```
00001110111000000011110111100000011110
11100000111101111000000111101111000000
11110111100000011111110000001111111100
00001111111111100011111111110001110111
11101110011101101101110011111111111100
01111111111100000111111111000
```

The Rules of the Game

- Get a secret word from the tutor table
- Draw it on a grid. (5 x 5, 10 x 10 or 15 x 15)
- Encode it into a binary string.
  - If it's bigger than 5 x 5 convert with **Run Length Encoding**
- Run to the tutor table, give them your encoded message.
- Get a message to decode from tutors.
- Decode and guess.
- Tutor checks that you've encode, decoded guessed correctly and awards points on your points sheet.

How to win

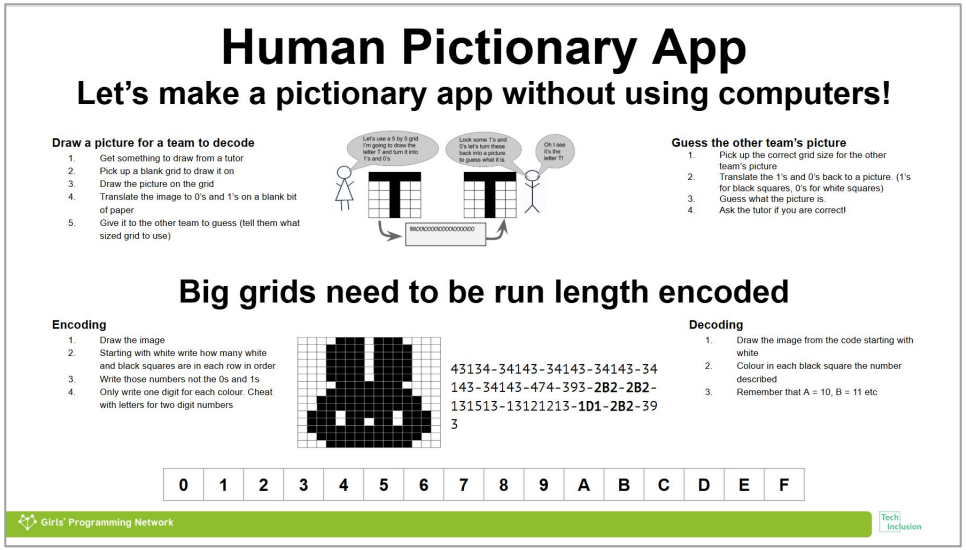
Here's the points table that will show you how to score yourself (keep track of these points!)

Action	5x5 Grid	10x10 Grid	15x15 Grid
Successful Encoding (correct binary)	5 points	10 points	15 points
Successful Decoding (correctly drawn)	5 points	10 points	15 points
Successful Guess (correctly identified)	10 points	10 points	10 points

Notice!! points are awarded for both **encoding** and **decoding** correctly**.**

Posters

If you forget any of those instructions posters are around the room to refer to.

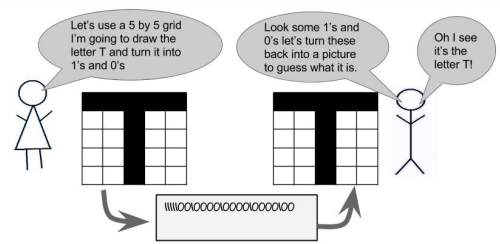


Human Pictionary App

Let's make a pictionary app without using computers!

Draw a picture for a team to decode

1. Get something to draw from a tutor
2. Pick up a blank grid to draw it on
3. Draw the picture on the grid
4. Translate the image to 0's and 1's on a blank bit of paper
5. Give it to the other team to guess (tell them what sized grid to use)



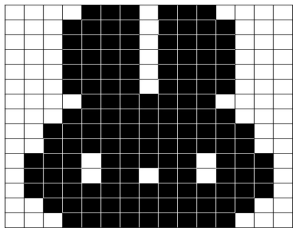
Guess the other team's picture

1. Pick up the correct grid size for the other team's picture
2. Translate the 1's and 0's back to a picture. (1's for black squares, 0's for white squares)
3. Guess what the picture is.
4. Ask the tutor if you are correct!

Big grids need to be run length encoded

Encoding

1. Draw the image
2. Starting with white write how many white and black squares are in each row in order
3. Write those numbers not the 0s and 1s
4. Only write one digit for each colour. Cheat with letters for two digit numbers



43134-34143-34143-34143-34143-34143-474-393-**2B2**-**2B2**-131513-13121213-**1D1**-**2B2**-393

Decoding

1. Draw the image from the code starting with white
2. Colour in each black square the number described
3. Remember that A = 10, B = 11 etc

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

The Rules of the Game

- Get a secret word from the tutor table
- Draw it on a grid. (5 x 5, 10 x 10 or 15 x 15)
- Encode it into a binary string.
  - If it's bigger than 5 x 5 convert with **Run Length Encoding**
- Go to the tutor table, give them your encoded message.
- Get a message to decode from tutors.
- Decode and guess.
- Tutor checks that you've encode, decoded guessed correctly and awards points on your points sheet.