Announcements

- When meeting to present your plan for your app (to be 'signed off'), you need to be able to explain its operation, and describe what the user sees – think of a "storyboard" from film
- Because finding time to work together can be a problem, getting the go-ahead early is smart

Arrays – Variables with Parts

Lawrence Snyder
University of Washington, Seattle

Arrays

- Arrays are variables naming multiple items of the same type rather than just one
  - In most programming languages, arrays are indicated by brackets [ ]
  - In Processing say: `int[ ] num = {2,4,6,8}` to define an array of four integers, initialized
- You've seen them before:
  - char[ ] codes = { 'S', 'R', 'L', 'U', 'D', 'H', 'A' }; // 7 elements
  - char[ ] team = { 'r', 'm', 'l', 'd'}; // 4 elements
  - String[ ] ninja, move; // Not defined
  - pixels[ ] // width*height // elements

Details

- Arrays are pretty intuitive

```
int[ ] coin = {1,5,10,25,50,100}
```

Index of element: 0 1 2 3 4 5

Reference elements with index in brackets

```
(coin[4] + 2 * coin[3]) ==  coin[5]
```

is true

Notice that the first item is indexed at zero

Properties

- The *length* of an array is the number of elements it has
- A handy way to refer to the length of, say, the `coin` array is:
  - `coin.length`
- The index of the last element is length - 1
  - `coin[coin.length - 1]`

  because the indexes start at `0`

Shouldn't it be "indices" rather than "indexes"? Either is OK … your choice

Datatypes

- Arrays can be of any datatype, just add brackets

```
int[] phoneNum = {2,0,6,5,4,3,2,1,0,0};
float[] const = {3.14159, 2.7182818};
boolean[] or = {false, true, true, true};
char[] song = {'Y', 'M', 'C', 'A'};
String[] preamble = {"We", "The", "People",
  "Of", "The", "United", "States", "Of",
  "America"} ;
color[] tempPixels = pixels[ ]; //must exist
```

It's even possible to have an array of fonts, but you must load them `PFont[ ] choice = new Pfont[3];`

Referencing Array Elements

- By saying the index of the element we want – called an *array reference* – we can use the elements just like normal variables.
- Examples

```
pixels[3]                       // numbers
pixels[i]                       // int variables
shade[col*width +row]           // expressions
shade[convert(row, col)]        // val returning
                                //    functions
```

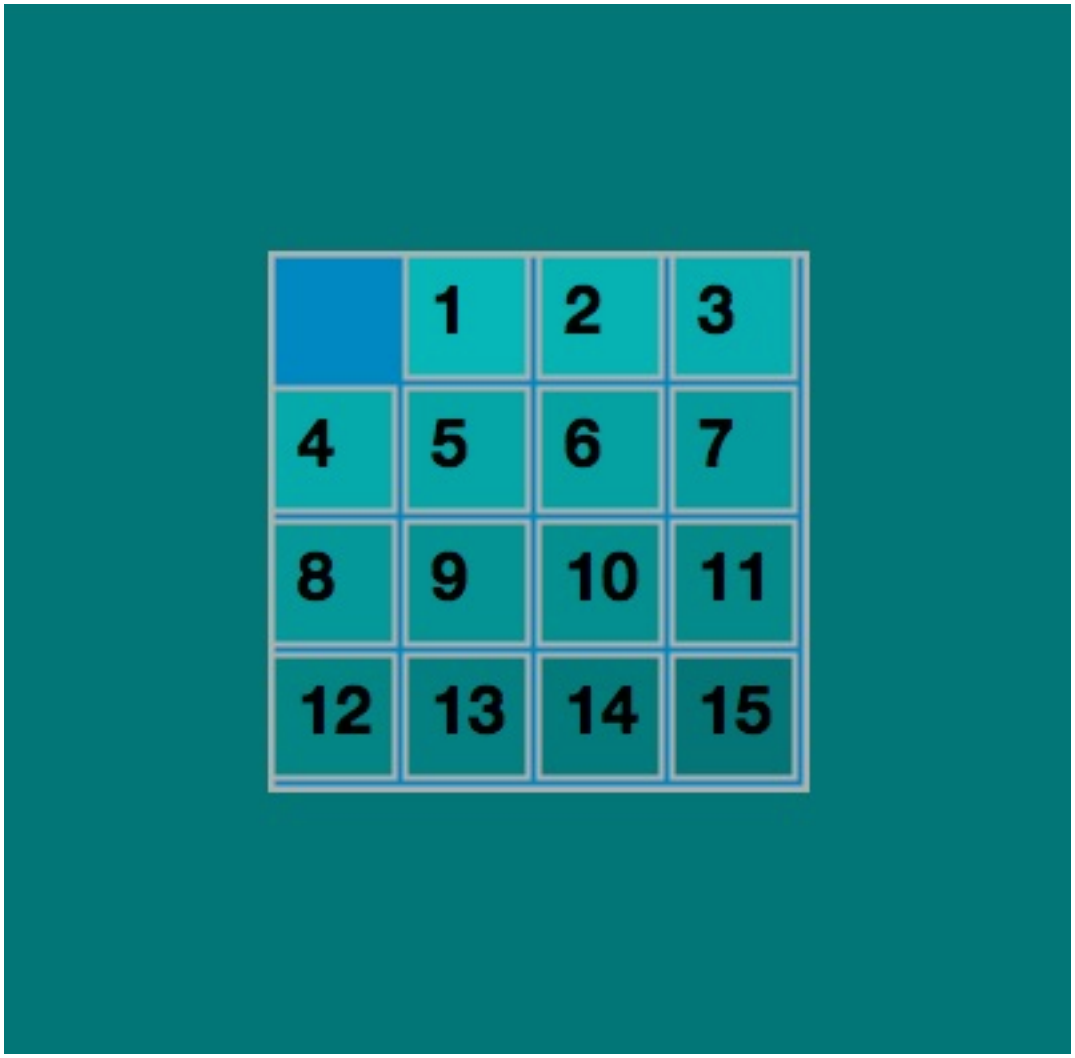
for-loops Reference All Elements

- We make most for-loops go from zero to n-1
- Those are the indexes values of arrays, too
- So, it's easy to reference all array elements to make a change

```
for (int j = 0; j < pixels.length; j++) {
   pixels[j] = color(0);
}
```

Working With Arrays … Little Surprise

The app works like the children's toy. To move a piece, click on the one to be moved



Basic Plan

- We draw the board by first drawing the game shape, then drawing each tile except the "`0`" tile, because it is the open space.
- Keep an array of length 16 containing the tile numbers; call it `pos` for position – it's 1-dimensional, like pixels[ ]
- Keep an array of length 16 containing the colors of the tiles – these will be progressively darker shades
- Need conversion function to go from 2-D to 1-D

Useful Facts

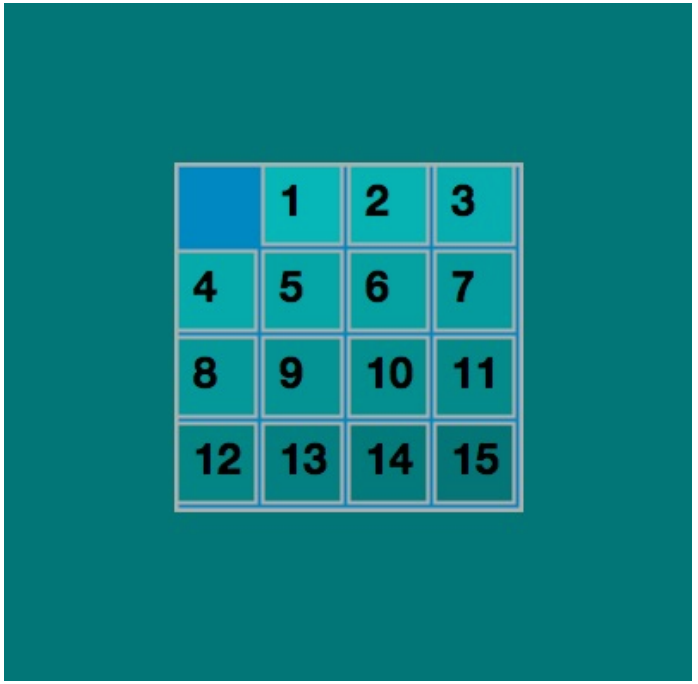
- Game starts at 100, 100
- Tiles positions are 50 x 50
- Colored part of tile is 46 x 46
- Need to compute: which tile the mouse clicked on … suppose `row` and `col` are `ints` then

```
row = (mouseY - 100)/50;
col = (mouseX - 100)/50;
```

- Must know if move is legal

```
abs(row-ex)+abs(col-wy)== 1
```

if `ex`, `wy` are row, column of `0`



Useful Facts

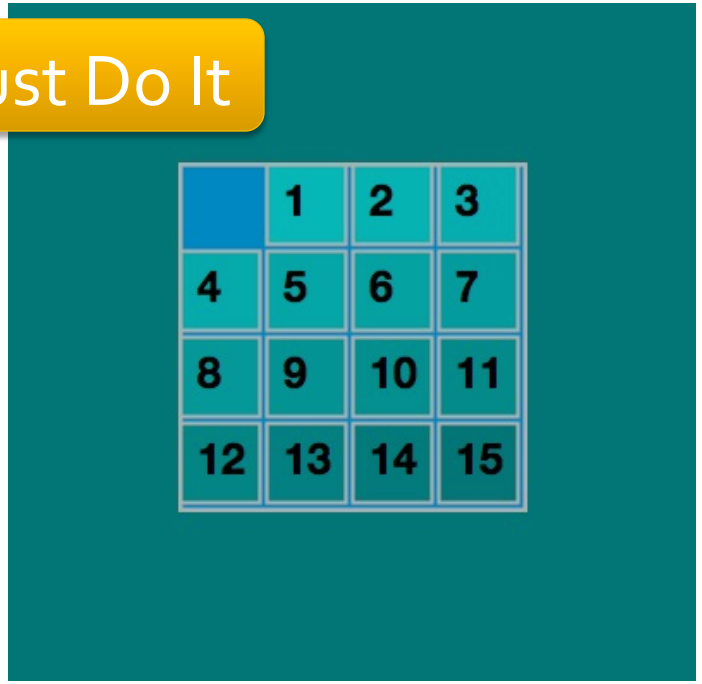
- Game starts at 100, 100
- Tiles positions are 50 x 50
- Colored part of tile is 46 x 46
- Need to compute: which tile the mouse clicked on … suppose `row` and `col` are `ints` then

```
row = (mouseY - 100)/50;
col = (mouseX - 100)/50;
```

- Must know if move is legal

```
abs(row-ex)+abs(col-wy)== 1
```

if `ex`, `wy` are row, column of `0`



Working With Arrays … Little Surprise

```
PFont digits;
int[ ] pos = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}; // tile vc
int ex=0, wy=0;                                       // position of the empt
int row, col, temp;                                   // tile position where
color[ ] shade = new color[16];                       // colors of each tile
color tempc;                                          // temporary for switch

void setup( ) {
  size(400,400);
  background(0, 100, 100);
  digits = loadFont("HelveticaNeue-Bold-25.vlw");
  textFont(digits);
  for(int i = 0; i < shade.length; i++) {
    shade[i] = color(15, 175-5*i, 175-5*i); // give the unique c
  }
}
```

Working With Arrays … Little Surprise

```
void draw( ) {
  fill(10,116,180);
  rect(100, 100, 200,200);                    // color background
  for(int i = 0; i < 4; i++){                 // loop through tile pos
    for(int j = 0; j < 4; j++) {
      if(pos[convert(i,j)] != 0) {            // if this isn't the emp
        stroke(150);                          // make tile edge medium
        strokeWeight(2);                      // use 2-wide line for t
        fill(shade[convert(i,j)]);            // locate tile color
        rect(102+50*j, 102+50*i, 46, 46);     // draw tile
        fill(0);                              // set number color to b
        text(pos[convert(i,j)], 102+50*j+10, 102+50*i+30); // place
      }
    }
  }
}
```

Working With Arrays … Little Surprise

```
int convert(int xpos, int ypos) {          // convert from 2-D ti
  return xpos*4 + ypos;
}

void mouseReleased( ) {
  row = (mouseY-100)/50;                                  // compute th
  col = (mouseX-100)/50;                                  // compute th
  if(abs(row-ex) + abs(col-wy) == 1) {                    // if the pos
    temp = pos[convert(ex, wy)];                          // use "3 ass
    pos[convert(ex,wy)]=pos[convert(row,col)];
    pos[convert(row, col)]=temp;
    tempc = shade[convert(ex, wy)];                       // use "3 ass
    shade[convert(ex,wy)]=shade[convert(row,col)];
    shade[convert(row, col)]=tempc;
    ex = row;                                              // remember
    wy = col;                                              //    is now
  }
}
```