JOEL’S LECTURES’ RECAP

1. Counting & remembering

> **Counting** (using groups of symbols to represent quantities/amounts)
* Decimal: ten symbols (0,1,2,3,4,5,6,7,8,9)
* Unary: only one symbol (1)
* Binary: two symbols (0 and 1)

> **Binary & computers**: computers use binary to store memory
* Bit: Binary digIT
* Byte: 8 bits
* Kilobyte: 1,024 bytes
* Megabyte: 1,048,756 bytes

> How **computers remember things** (through storing data as constants and variables) and change them (through commands)
* Data types: different values/types of storing numbers and words in a computer. They can be variable (value can change) or constant (value is fixed).
  - Whole number/Integers: store as ints
  - Numbers between whole numbers: store as floating points/floats
  - Individual characters of alphabet: store as chars
    > There’re different encoding like ASCII and Unicode
  - Multiple words (multiple characters): store as strings

> **JavaScript**: programming language
* All lines must finish with ‘;’
* Keyword ‘let’ used to create new data/variables
* Commands/functions allow to do things to data

2. Drawing & colouring

> **P5.js:** open source JavaScript library for creative coding. Based on Processing.
* P5 reference
* P5 web editor: online platform for learning to code in a visual way

> **Functions:** take an input and give an output

> **Dimensions:** could be though of as individual variables
Example of dimensions in drawing:
* Zero dimensions: a point
* One dimension: a line
* Two dimensions: a square
* Three dimensions: a cube

> **Drawing:** need a way to say where we are in screen and how to draw shapes
* (x,y) pairs used to describe Cartesian co-ordinates.
  - X: how much left or right
  - Y: how much up or down
* The value 0 of Y in computers is on top left corner because computers scanned from top to bottom using Cathode Ray tubes
* Shapes
  - Circles: set of all points in a (flat)plane that are at a given distance from a given point.
  - Sine waves: curves that go up and down in a regular way forever
* Shape primitives in p5.js: triangles(), rect(), quad(), ellipse(), arc()
* Transformations (move drawing tool around)in p5.js: translate(), push(), pop(), rotate(), scale()

* Voronoi diagram: diagram of boundaries between a set of points such that within the boundaries, one is closer to that point than any other.

> **Colouring**

* Digital colour mixing is additive
* Complementary colours: opposite sides of colour wheel
* RGB: mixing Red, Green, Blue values to make any other colour
* HSB: mixing Hue, Saturation, Brightness values to make any other colour

* Colour in digital: defined as a range of numbers
    - Black & White: 0 is black, 255 is white, in between is a shade of grey
    - Colour: mixing RGB or HSB values between 0 and 255. Can also include another value for transparency: alpha.

3. Deciding

> **Logic:** computers use it to decide what to do

* Control flow: how you control the flow of which part of the programme is executed next.
    - Comparisons are used in control flow statements: >=, <=, ==, !=, ===, !==
* Boolean data type: two values - true(1) or false(0)
    - Boolean operators - combine or invert boolean data types : ||, &&, !
* Conditional statements: allow program to execute a block of code, based on the result of an expression that utilizes relational or logical (boolean) operators
    - If/Else statements
    - Switch statements
* Scope: part of the program where a name binding (an association of a name to an entity, such as a variable) is valid
    - Global scope: data storage defined outside of a function.
    - Local scope: data storage defined inside of a function.
* Some functions in p5.js: noLoop(), mousePressed(), redraw()

> **Functions:** allow programmers to package up functionality into smaller mini programs. Functions take input and return output

* Recursion: method of solving a problem where the solution depends on solutions to smaller instances of the same problem.
    - Fractals

> **Computing history**

* The Jacquard Loom: worlds first programmable object
* Charles Babbage: inventor of the concept of a digital computer. Invented Difference Engine and Analytical Engine
* Ada Lovelace: worlds first computer programmer
* Panini: pioneered a grammar that incorporated metarules and recursion - made first formal language
* Muhammad ibn Musa al-Khwarizmi: formalised algebra, his name gave rise to the word algorithm and introduced the decimal system to the western world.
* Ismail al-Jazari: invented programmable machines
* Alan Turing: formalised the concept of the algorithm and computation in general
* Grace Hopper: invented the compiler
* Margaret Hamilton: led the development of on-board flight software for NASAs Apollo Moon missions.

> **Three dimensions**

* Third dimension: described in terms of the letter ‘z’
* Storing three dimensional information:
    - Point clouds: set of data points in space

- Voxels: represents a value on a regular grid in three-dimensional space
- Polygons: plane figures (flat) with at least three straight lines and angles.
  > Examples: squares, triangles, pentagons
- Polyhedron: three dimensional shape made of multiple polygons.
  > Platonic solids: type of polyhedron which are constructed by congruent (identical in shape and size), regular (all angles equal and all sides equal), polygonal faces with the same number of faces meeting at each vertex. Five solids meet these criteria: tetrahedron, cube, octahedron, dodecahedron, icosahedron

4. Repeating

> **Iteration:** repetition of a process in order to generate an outcome

> **Loops**: part of the control flow. A loop is a sequence of statements which is specified once but which may be carried out several times in succession.

* While loops: code to be executed repeatedly based on a given Boolean condition.
* For loops: code to be executed repeatedly.
* Loops in P5.js
  - for: loops through a block of code a specified number of times

```
for (statement 1; statement 2; statement 3){
code block to be executed
}
```

  - for/in: loops through the properties of an object
  - while: loops through a block of code while a specified condition is true
  - do/while: variation of while loop, loops through a block of code while a specified condition is true

5. Listing & randomising

> **JavaScript objects**: used to store keyed collections of various data and more complex entities.

* Created with figure brackets {…}
* Can contain optional list of properties (a property is a “key: value” pair)
  - Key: string (also called a “property name”)
  - Value: anything
* Property values are accessible using the dot notation

> **Arrays**: data structure to store ordered collections

* Created with square brackets […]
* Array elements are numbered, starting with zero
* Methods pop/push, shift/unshift
* A strings is a one dimensional array of chars
* Two dimensional arrays: storing an array within an array
  - Images are suited to be stored in 2D arrays
  - To store 2D array in 1D: put each row of the 2D array in a long row in the 1D array
* N (or any) dimensional arrays: storing multiple arrays within arrays

> **Randomness**

* P5 random() function: returns a random floating-point number.
  - Only produces pseudo-random numbers.
  - For real randomness is best to sample nature
* Noise
  - Perlin noise: type of noise invented by Ken Perlin in 1983. Used to generate realistic looking clouds, landscapes or many other natural features
    > P5 noise(): returns the Perlin noise value at specified coordinates
  - Simplex noise: *n*-dimensional noise

- Worley noise: noise to produce procedural textures
- Colours of noise: power spectrum of a noise signal

6. Timing

> **P5 draw**: draw() functions runs as a loop. The code inside it runs continuously from top to bottom until the program is stopped.

* draw() loop may be stopped by calling noLoop() and can be resumed with loop()
* If noLoop() inserted in setup(): should be the last line inside the block, and it'll make the code within draw() only run once at the start of the program
* redraw() function: executes the code within draw() one time. Allows the program to update the display window only when necessary

> **P5 events**

* Some mouse events: mouseX(), mouseY(), mouseIsPressed(), mouseClicked(), mouseDragged(), pmouseX(), pmouseY()…
  - Arrays can be used to store mouse interaction information over time
  - A vector can be used to store mouseX and mouseY in one object
* Some key events: key(), keyIsPressed(), keyTyped()…

> **Linear interpolation (blend between values) in P5**

* lerp() function: calculates a number between two numbers at a specific increment
  - lerp(start, stop, amt)

* map() function: re-maps a number from one range to another
  - map(value, start1, stop1, start2, stop2, [withinBounds])

* Easing: process of making an animation less severe or pronounced so it looks more natural

> **Time in P5:** year(), month(), day(), hour(), minute(), second() functions return the current value of them

> **Frames in P5**

***** frameCount() function contains the number of frames that have been displayed since the program started. It depends on frameRate() function

* deltaTime() variable: contains the time difference between the beginning of the previous frame and the beginning of the current frame in milliseconds regardless of frame rate
* millis(): returns the number of milliseconds (thousandths of a second) since starting the sketch (when `setup()` is called)

> **Asynchronous programming**: occurrence of events independent of the main program flow

* Asynchronicity in P5: preload() function forces the program to wait until everything has loaded before moving on. Useful to preload images, videos, audios etc.

* Scheduling: executing a function not right now, but at a certain time later. Methods:
  - setTimeout() allows us to run a function once after the interval of time
  - setInterval() allows us to run a function repeatedly, starting after the interval of time, then repeating continuously at that interval. Will continue calling the function until clearInterval() is called

* Callbacks: second argument is a function (usually anonymous) that runs when the action is completed.
  - Handling errors: if there's an error the callback should be able to react on that. A callback function should contain an error argument if it occurs

* Promises: allow you to defer further actions until after a previous action has completed, or respond to its failure. It's an object that represents an intermediate state of an operation — in effect, a *promise* that a result of some kind will be returned at some point in the future

7. Classing

> **Some computing pioneers**

* Seymour Papert: interested by learning theories - especially how computers could help children to learn. Pioneered the idea of microwords

* Alan Kay: pioneered thinking around Object Orientated programming and Overlapping Window Graphical User Interfaces

> **Object oriented programming (OOP):** programming based on the concept of "objects", which can contain data and code

* Through OOP is possible to create a library: a collection of related pieces of **code** that have been compiled and stored together in a single file

  - Libraries of code can be abstracted into API (Application Programming Interfaces): way of abstracting a load of complexity behind a series of commands and formats of data

* Abstraction: key concept in OOP. Allows you to combine logic and data into one place.

* Class: code that combines data and methods (functions) in to one place for one object. To make a variable (or object) out of a class you need to instantiate it with the new command. An object is an instance of a class.

* Encapsulation: an attribute of object design. It means that all of the object's data is contained and hidden in the object and access to it is restricted to members of that class.

> **Libraries:** collection of related pieces of **code** that have been compiled and stored together in a single file

* P5.js is a library

* To use a library, you have to "install" it in your sketch by downloading and linking it in your js and html files

*Examples of libraries

  - Gui.js: generates a graphical user interface
  - Matter.js: 2D physics engine for the web
  - Moment.js: helps to deal with timezones, dates and times
  - Paper.js: vector graphics library
  - Two.js: vector graphics library
  - LiquidFun: liquid simulation library
  - Zdog: pseudo-3D engine
  - Pts: linear algebra library
  - Three.js: 3D graphics library
  - Hammer.js: touch gesture library

8. Images and simulation

> **Four dimensions:** mathematical extension of the concept of three-dimensional or 3D space

* Examples from 3D to 4D: cube to hypercube, sphere to hypersphere etc

*Time can be thought as the fourth dimension

> **Artists working with space and time**

* Eadweard Muybridge, photographer important for his pioneering work in photographic studies of motion.

- ***** Étienne-Jules Marey: scientist who invented the photographic gun.
- ***** Marcel Duchamp: artist associated with Cubism, Dada and conceptual art
- ***** Umberto Boccioni: artist who helped shaped the Futurism movement
- * James Turell: artist known for his work within the Light and Space movement
- * Richard Long: sculptor and land artist

> **Images in P5**

***** p5.Image: data type to store images in P5. It's a canvas backed representation of an image.

- createImage() function (same as new P5.image): creates a new P5.Image
- .pixels: gives access to an array containing the values for all the pixels in the display window. These values are numbers
- Before accessing the pixels of an image, the data must loaded with the loadPixels() function. After the array data has been modified, the updatePixels() function must be run to update the changes.

***** image(): draws an image to the p5.js canvas.

* loadImage(): **l**oads an image from a path and creates a p5.Image from it.

> **Compressions**

*JPEG: spatial compression for still images

*MPEG: temporal compression for moving images and sound

*Glitch: short-lived fault in a system, such as a transient fault that corrects itself, making it difficult to troubleshoot

> **Fast Fourier transform:** algorithm which splits any waveform into a series of simpler waves

> **Feedback:** when you feed the output of a signal into the input of it

***** Audio feedback

* Video feedback

> **Simulations**

***** Cellula automaton: model of computation studied in automata theory.

- Conway's game of life: cellula automaton which simulates biological life with four rules. Invented by John Conway in 1970
- Nils Aall Barricelli: pioneer on the field of Cellula Automata
- Stephen Wolfram: computer scientist studied cellula automata in 1980s

***** Boids: artifical life system developed by Craig Reynolds in 1986. It simulated realistic flocking with just three simple rules. Example of emergence (the whole is greater than the sum of the parts)