HARDWARE / SOFTWARE PRESENTATION

SOFTWARE DESIGN & DEVELOPMENT 2016

Sound Processing
Sound Editing
Lua

RYAN OO

JAI RAINA

ANDREW WONG

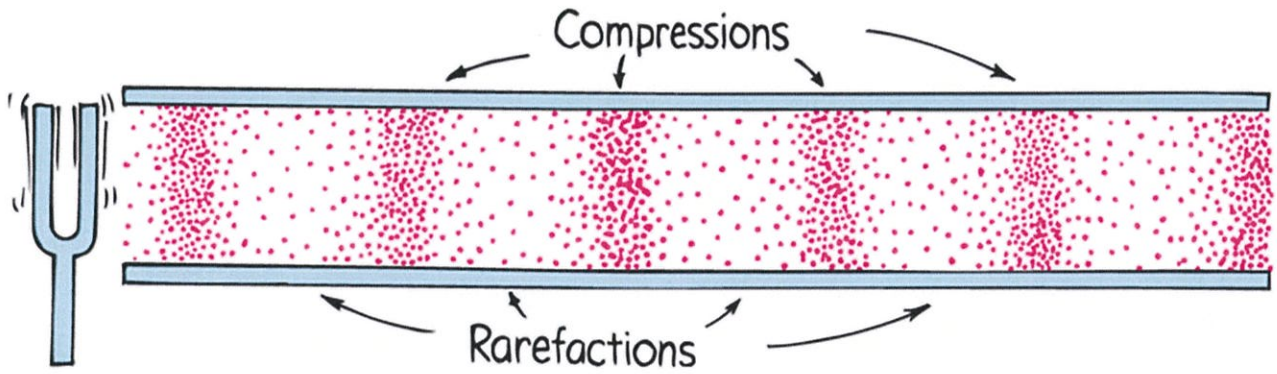
Ryan Oo, Jai Raina, Andrew Wong

Hardware Category: Sound Processing

Overview

Analogue and Digital Waves

The sound that we hear is analogue sound. It is a series of compressions and rarefactions in a medium, in this case air. Our ear receives the compression and rarefactions and sends electrical signals to our brain which interprets it as sound. However, a method of capturing sound needs to be made for computers, as they use binary. To do this, we have to transfer analogue waves to digital waves. A microphone does this. There are many types of microphones, but the two most common ones are condenser and dynamic microphones.



Types of Microphones

Condenser microphones have a diaphragm, which is considered the front plate of the microphone. The diaphragm vibrates in and out in response to the compressions and rarefactions of the soundwave. A back plate behind the diaphragm stays still. An electric current flows between the diaphragm and back plate. As the diaphragm moves closer to the back plate in response to the soundwave, the electrical current flows more freely to the back plate. As the diaphragm moves further away from the back plate, the current flow is reduced. Thus, the electric current represents the changes in the sound wave. The electric current is an analogue signal of the sound wave, and is sent to an ADC (analogue to digital converter), which transforms the analogue signal into digital data (binary).

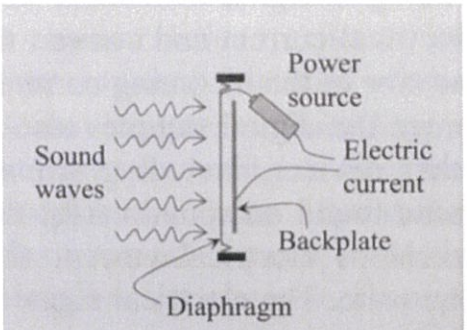


Fig 2.21
Detail of a condenser microphone.

Dynamic microphones have three important components, a diaphragm, a copper coil of wire and a magnet. The diaphragm is attached to the copper coil of wire, and the copper coil of wire is surrounded by the magnet. As the diaphragm moves in and out in response to the sound wave, so does the copper wire. This causes the copper wire to create an electric current with the magnetic field. The analogue signal (electrical current) in the copper wire is sent to an ADC, where it transforms the electrical current into digital data.

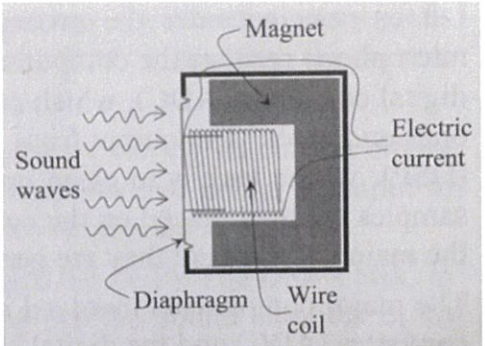


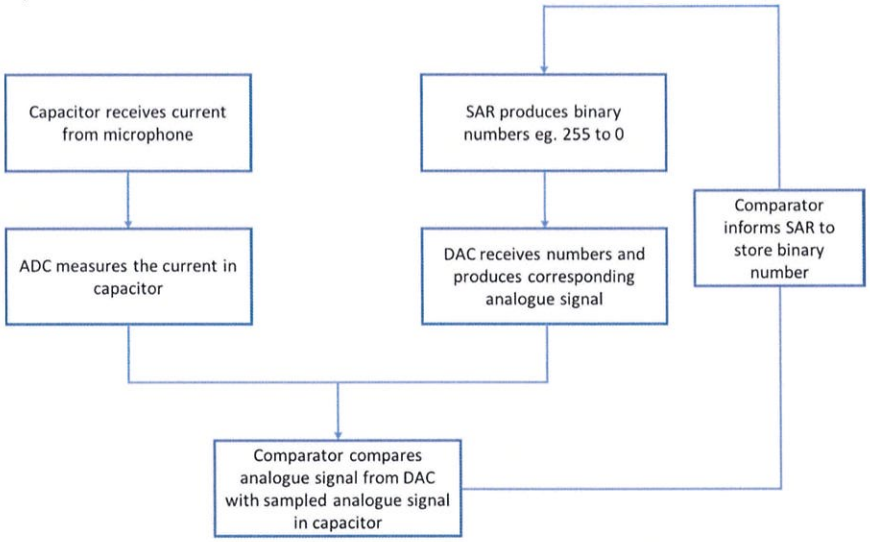
Fig 2.20
Detail of a dynamic microphone.

Ryan Oo, Jai Raina, Andrew Wong

How It Works

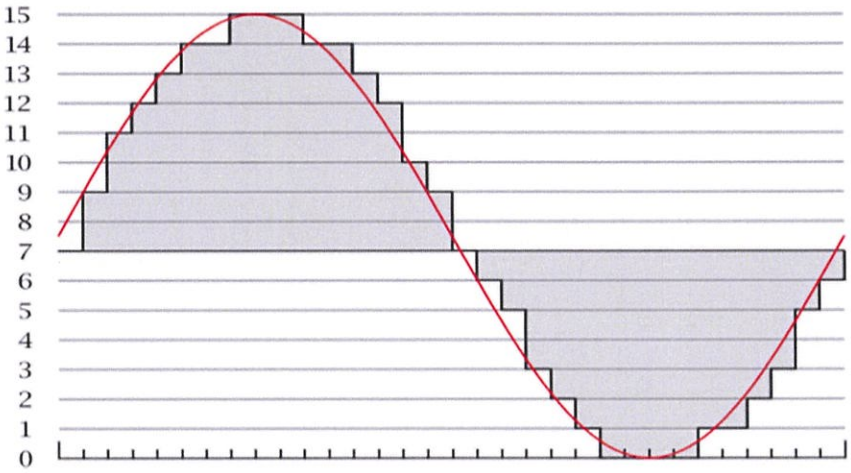
Sound Sampling

There are five devices that allow sound sampling to take place; ADCs (analogue to digital converters), DACs (digital to analogue converters), SARs (successive approximation register), DSPs (digital signal processer) and comparators. The analogue signal from the microphone is fed into a capacitor at precise intervals eg. 44100Hz. The ADC measures the current of the analogue signal in the capacitor. A successive approximation register (SAR) produces binary numbers eg. 255 all the way down to 0. The DAC receives these numbers and produces the corresponding analogue signal. A device called a comparator compares the analogue signal from the DAC with the sampled analogue signal in the capacitor, if it is the same analogue signal, the comparator informs the SAR, and the SAR stores the number generated.



This number represents a sample of the ***amplitude*** of the analogue wave captured by the microphone and is converted into a series of ones and zeroes to represent binary data. This process is repeated multiple times a second, eg. at the rate of 22,000Hz and 44,100Hz, which is the speed of sampling achieved with modern technology. An audio file sampled at a rate of 22,000Hz will have less quality than an audio file sampled at 44,100Hz, but it would be half the file size of the 44,100Hz sound sample. The sound samples are then sent to a DSP (digital signal processor), which can perform a variety of tasks on the sampled sound, including wave shaping, to make the transitions from sample to sample smoother. DSPs can also remove background noise from music, as background noise has different characteristics than music.

To reproduce the sampled sound, the amplitude values captured by the sound sampling is converted into an electrical current. High amplitude = High current, low amplitude = low current. The speaker contains a diaphragm which is connected to a copper coil of wire. The copper coil of wire is surrounded by a magnet. When the electrical current is applied to the copper coil, it causes the magnetic field to fluctuate, vibrating the copper coil in response to the changing of the magnetic fields. This causes the diaphragm to also move, causing compressions and rarefactions in the air we hear as sound.



Historical Development

Acoustic Era (1877 to 1925)

The quality of sound captured has vastly improved over the years, as has many aspects of technology. The earliest form of recording sound was called the acoustic era. A phonograph was used to record sound during this era and it had a large horn that collected the sound waves around it. A diaphragm was connected to the apex of the horn, that moved according to the compressions and rarefactions in the sound wave. The diaphragm was connected to a scribe, and as the diaphragm moved in response to the soundwaves picked up by the horn, the scribe scratched the surface of a soft metal to record an analogue of the soundwave on the surface of the soft metal. This method of recording sound was only able to capture sounds of frequencies between 250Hz and 2,500Hz and only allowed a low quality of sound to be recorded.

Electrical Era (1925 to 1945)

This was the second era of sound recording, and employed the use of electronic microphones to collect the sound, electronic amplifiers to strengthen the sound and electronic disc cutters to record the sound on metal discs. Condenser and dynamic microphones were used to transform analogue sound into analogue electrical signals. These signals were fed to an electronic amplifier which strengthened the signal, making the analogue sound louder. The signal then went to an electronic disc cutter, which transformed the electrical signals into mechanical movement, scribing the analogue onto a polyvinyl plastic disc. This new improvement in sound recording allowed frequencies of 60Hz to 6,000Hz to be recorded, and improved sound fidelity.

Ryan Oo, Jai Raina, Andrew Wong

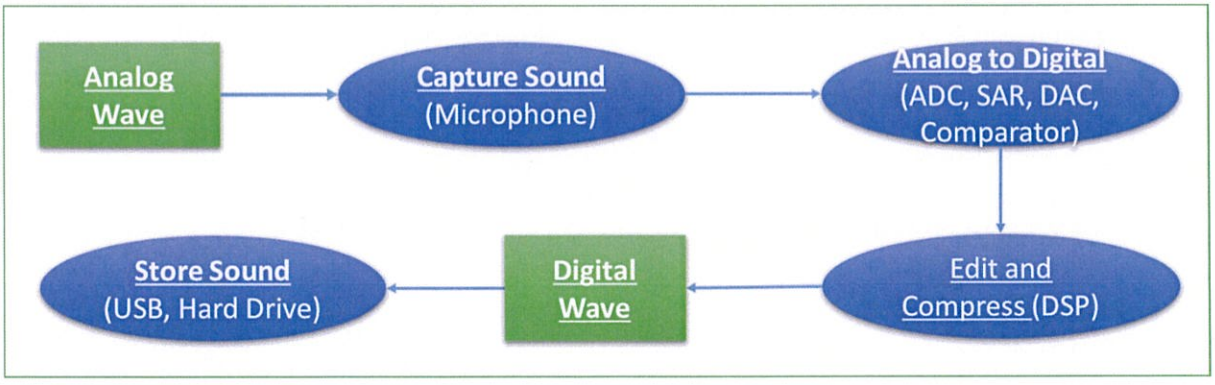
Magnetic Era (1945 to 1975)

The third era of sound recording involved the use of magnetic tape recorders. A strip of magnetic material moves at a constant speed past an electronic write head. An analogue electrical signal captured by a microphone is fed to the write head. The write head passes the electrical signal to the magnetic strip running past the head, inducing a pattern of magnetisation similar to the signal in the write head. This method of storing sounds greatly increased the fidelity of the sound recorded, and also allowed the storage medium to be more portable, as the method of storage is more efficient.

Digital Era (1975 to now)

The fourth and current era of sound recording captures a discrete signal many thousands of time a second eg. 44,100Hz. An analogue signal from a microphone is transferred into binary data by a variety of means (indicated previously in the seminar notes), and can be stored on a variety of devices eg. USBs, Hard Drives and CDs. USBs use a method of electrically storing recorded sound, Hard Drives use a magnetic method of storing recorded sound and CDs use an optical method of storing sound. Digitally stored sound are able to capture sound of the highest and most accurate fidelity and are able to produce frequencies 20Hz to 20,000Hz (the sound range that humans are able to hear). Digitally stored sound is even capable of storing sound in more frequencies than humans can hear, but it would not be practical. Along with this, digitally stored sound is extremely portable, as it can be carried in your pocket, in the case of USBs and can fit in your hand, in the case of CDs.

Journey of Analogue Sound to Digital Storage

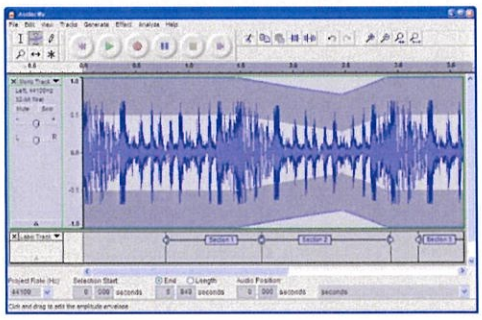


Ryan Oo, Jai Raina, Andrew Wong

Software Category: Sound Editing

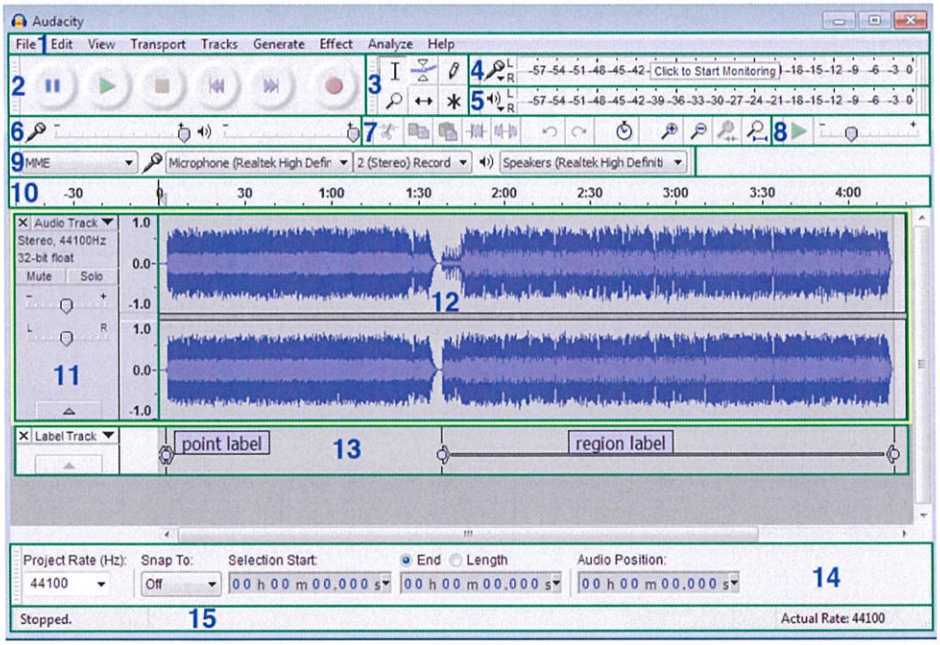
Overview

Sound editing has been present in all of society for decades; mainly through the entertainment industry. Movies, television, games and of course music require sound editing. Not only do musicians or filmmakers necessitate over sound editing; but students have potential use for it as well for school assignments, projects or for recreational leisure. Sound editing has been achieved in the past and the present through 2 key ways. Analogue editing and digital editing. Analogue editing is a traditional tape-based post-production and in contrast to digital editing which uses software based tools on a computer. A few recognisable examples of digital sound editing software include: Audacity, Wavepad, FL Studio, Adobe Audition, just to name a few. For the rest of the 'Sound Editing' portion of this seminar, the presentation will be focused and based on the sound editing software known as Audacity.



How It Works

Audacity is a multi-track audio editor and recorder which includes a variety of features which help users in creating and editing audio files. Below and to the right is a guide on how to use and navigate Audacity.



1. **Menu Bar –** Offers commands such as: file, edit, view, transport, tracks, generate, effect, analyse, and help.
2. **Transport Toolbar** – Provides buttons for controlling playback, recording and moving the project to the start or the end; all of which are achieved by the pause, play, stop, skip to start, skip to end and record buttons.
3. **Tools Toolbar** – Provides buttons known as:

- Selection (which can be used to mark a starting point for audio playback);
- Envelope (allows volume changes)
- Draw (which can also be used to change volume),
- Zoom (which can be used to zoom in and out of the audio track to see greater detail of certain sections)
- Time shift (allows to synchronize audio in a project by dragging individual or multiple audio tracks, note tracks or audio clips left or right along the 'Timeline')
- Multi tool (allows access to all of the five separate tools on the toolbar at once.)

4. **Recording Meter Toolbar –** Displays audio being recorded
5. **Playback Meter Toolbar –** Displays audio being played
6. **Mixer Toolbar –** Changes recording volume (on the left) or the playback volume (on the right)
7. **Edit Toolbar -** Provides buttons such as: cut, copy, paste, trim audio, silence audio, undo, redo, zoom in, zoom out, zoom selection, and fit project; all of which are self-explanatory. However the Sync-Lock Tracks button requires a little explaining. The button ensures that length changes occurring anywhere in a defined group of tracks also take place in all audio or label tracks in that group, even if those tracks were not selected. This lets you keep existing audio synchronized with each other.
8. **Transcription Toolbar –** Allows users to play audio at slower or faster speeds than normal.
9. **Device Toolbar –** Helps in selecting: audio host, recording device, recording channels and playback device
10. **Timeline –** Displays a horizontal ruler above the tracks measuring time from zero (the start of the track)

11. **Track Control Panel** – It is actually a part of the audio track which Includes controls to change the track. Controls include:

- Close Button – Closes the track, removing it from the user's project
- If the user clicks the Audio Track Dropdown Menu, more controls are provided such as the:
- Mute button which silences the track selected
- Solo button which plays the single track selected
- Gain slider which sets the gain for track selected
- Pan Slider which makes signal stronger on left or right earphone

There are also status indicators which provides information on the track: showing if it is stereo or mono; the sample rate (Hz); and the sample format.

12. **Audio Track** – Shows the users a Vertical Scale with units. This scale displays amplitude when showing the waveform, or frequency when showing the spectrum or pitch. The audio track also provides a visual representation of the audio waveform. Tracks like these are created whenever a new audio file is recorded or imported.
13. **Label Track** – This is an additional track which can be added into a user's project. Labels are used reference points or regions in an audio track. However the label track itself does not hold audio. In addition, labels can also be used to contain text to annotate a certain part of the track.
14. **Selection Toolbar** – This includes controls for 'Project Rate' (sample rate), 'Snap To' and for manipulating units of time or other selection formats.
15. **Status Bar** – This bar is always at the bottom of the page, and its core function is to display messages about recording or playback; indicate parts of Audacity as well as provide hints on how to use Audacity for the user.

Main Editing Features of Audacity

Audacity provides great features for users to work with. Examples of these great features include:

- Simple manipulation and editing using the cut, copy and paste buttons
- Unlimited sequential undo and redo, giving users more flexibility; encouraging them to try new things.
- Mixing and compiling as many tracks as users need which allows freedom for creativity. This technique can be completed using 2 ways. The first procedure is called 'Mix and Render' and the second one is called 'Mix and Render to New Track'. Mix and Render simply mixes down any selected tracks; to a single mono or stereo track, rendering to the waveform all real-time transformations that had been applied. The resulting track also known as the 'Mix' track replaces the selected tracks and is placed underneath any tracks that were not mixed and rendered. The other technique, 'Mix and Render to New Track' is almost the same thing. The one major difference being the original tracks are preserved rather than being replaced by the 'Mix' track.
- Putting in unique sounding effects or in contrast reducing sound; depending on what the user is trying to achieve. This can be accomplished by clicking the effect button, where the user can do a variety of things, for example: change the tempo, change the pitch, reduce noise, add in an echoing effect or amplify the audio.

Ryan Oo, Jai Raina, Andrew Wong

Historical Development

Audacity, the audio editing software was first worked on by Dominic Mazzoni and Roger Dannenberg in 1999, as a research project for university. It was first released on the 28th of May, 2000 as version 0.8. By June, 2002 they released version Audacity 1.0. Near the end of 2003, Audacity released the 1.2 series, which introduced hundreds of new features such as; support for professional-quality 24-bit and 32-bit audio, a comprehensive manual, and it provided non-English speaking languages supporting other countries. After the run of the 1.2 series, a Beta 1.3 series with enhanced features was released which serves as a foundation for the current 2.0 series for Audacity. This Beta that was released had new features such as: new keyboard commands, new shortcuts, the 'Snap To' button was introduced, an enhanced interface, new effects, and overall improvements towards the whole software. As more versions were slowly being released, incremental changes were being applied to Audacity. With each new version came more bug fixes, improved interfaces, new effects, new files could be imported and exported. By version 2.0, Audacity was highly enhanced and improved from its predecessors. During the 2.0 series (which is still running today) creativity and freedom for users is one of the main goals for Audacity. Giving users expansive boundaries to go towards. Audacity is now smoother, more effective and has more variety for users to work with. Audacity is now available for Windows, Mac, OS X, Linux, and offers over 30 languages for users.

Ryan Oo, Jai Raina, Andrew Wong

Programming Language: Lua

Overview

Lua is a cross-platform, extension programming language that appeared 23 years ago in 1993 – publically in 1994. It was created to serve as a lightweight scripting language that could extend and customise the functionality of programs. In other words, the Lua programming language was not the primary programming language of a program, but rather a side language, which helped increase development efficiency.

Lua was developed in the Pontifical Catholic University, by three people: Roberto Ierusalimschy *(Ee êh roo zah leems key)*, Luiz Henrique de Figueiredo, and Waldemar Celes. During 1977 till 1992, Brazil has a strict market reserve (aka trade barrier) that made it difficult for them, as members of their university's Computer Graphics Technology Group (Tecgraf) to buy nor sell custom computer software. So, in response? They built their own software.

Lua is a programming language very similar to other languages, and should be easy to pick up on. It has the dynamic scripting feel like JavaScript, where the outcomes of code changes can easily be seen, and also has a similar code syntax with C and Pascal, two other programming languages.

Fun fact: the name of this programming language, Lua, is the Portuguese word for Moon. This programming language was built as a successor of a previous language, called SOL (Simple Object Language). As SOL meant Sun in Portuguese, the creators of Lua decided to name their programming language the opposite of Sol, as a joke.

Fun fact #2: the programming language was awarded the 2011 Game Developer Magazine Front Line Award

How It Works

Creating Lua Scripts

Lua scripts are simply plain-text files, and hence can be written with any text editor. However, special editors, known as Integrated Development Environments (IDEs) have features that assist the developer, such as code completion, syntax highlighting, error checking, and etcetera.

Lua scripts can be identified if their file extension are ".lua"

Programming Syntax

Like many other programming languages, Lua is not strict on the formatting of the code itself – spaces, line breaks and indentations do not affect the operation of the software. Dreaded by some, the semicolon, used to indicate the end of a line, is optional.

At the basic level of Lua, it is also able to do mathematical calculations and comparisons. Their syntax is shown below

`+`	add	`1 + 2`	`3`
`-`	subtract	`9 - 4`	`5`
`*`	multiply	`2 * 3`	`6`
`/`	divide	`9 / 9`	`1`
`%`	modulo (remainder after division)	`9 / 6`	`3`
`^`	power	`2 ^ 2`	`4`
`==`	is equal to	`9 == 2`	`false`
`~=`	is not equal to	`9 ~= 2`	`true`
`<`	is less than	`9 < 2`	`false`
`>`	is greater than	`9 > 2`	`true`
`<=`	is less than or equal to	`5 <= 6`	`true`
`>=`	is greater than or equal to	`5 >= 5`	`true`
`#`	length of	`#{1,2,3}`	`3`
`--`	comment (until end of line)	`... --comment`	
`--[[`	open comment block	`... --[[ comment block`	
`--]]`	close comment block	`spam spam spam --]]`	

`0.5`	number
`1`	number
`"string"`	string
`'string'`	string
`\`	escape character
`nil`	undefined / null

Syntax/Language Comparison

- End of line semicolons are optional
- Braces { } are replaced with `do` and `end`.
- Arrays are 1-based (array position starts from 1)
  `print(({"A","B","C"})[0])` nil
  `print(({"A","B","C"})[1])` A
- No ternary operator (`a?b:c`) – aka no 'one liners'
- Multiple assignment `x,y = y,x`
- Numbers are numbers (No difference between integers and floats)
- Undefined/null values are represented as `nil`

Variables

Variable Types

Variables are data items which store information. In Lua, also known as an identifier, a variable can take form as a global variable or a local variable (or a variable in a table – a property)
A global variable is accessible at any point of the code. It can be declared by `variable = 10`
A local variable can only be accessed by the functions in the same nesting, and can be declared by appending the keyword `local` before the variable declaration. `local var = 3`

If a variable is referenced without being declared, it will be undefined, or `nil`

Variable Naming Conventions

Variables can be named anything, provided that they:

- Are not the word as Lua's programming keywords
- Contain only alphanumeric and underscore characters (A-Z, a-z, 0-9, _)
- Do not begin with a digit

Memory Management

In context of software development, programmers need to be aware of how much memory their software is consuming. If their code is laden with too many unused variables, or has a memory bug, the application may freeze and crash, or even worse, crash the whole computer. To stop these memory leaks, the programmer must be aware of how much memory their application is using, and take necessary memory management precautions

The Lua programming language has inbuilt automatic memory management, also known as Garbage Collection (GC). At regular intervals, the program will automatically free up memory by releasing variables that will no longer be used. Garbage Collection, however, does not affect global variables, nor properties in a table.

To delete variables, simply set the variable to a value of `nil`

Advantages Over Other Languages

- Easier use for prototyping compared to other languages
- Shorter development time (compared to C++ for example)
- Compilation can occur at runtime (code does not have to be manually compiled)
- Lightweight (not too much memory overhead)
- Relatively simple syntax – very close to pseudocode

Historical Development

Lua has undergone many updates, and is currently at version 5.3.2, which was released in November 2015. Each iteration brought new features, functions, bug fixes (and bugs...), more efficient code execution, and support to more and more devices.

Applications

As mentioned earlier, Lua is not intended as the primary programming language of an application (albeit possible). Instead, Lua is embedded in a program with another main language. It is commonly used in games for scripts that are triggered in a level.

Well known software/games/whatever that use Lua include: the VLC player, TeamSpeak, Garry's Mod, CryEngine, Adobe Lightroom, World of Warcraft, and many more. It also can be used to script the LEGO Mindstorms NXT unit.

Bibliography

Software Design and Development Preliminary Course – Samuel Davis. Page 61

https://en.wikipedia.org/wiki/Sampling_(signal_processing)

http://www.tech-faq.com/how-does-usb-work.html

https://en.wikipedia.org/wiki/History_of_sound_recording

http://ethw.org/Electrical_Recording

http://www.lloydmicrophoneclassics.com/mic_history.html

http://computer.howstuffworks.com/flash-memory1.htm

http://www.g-w.com/pdf/sampchap/9781590707678_ch24.pdf

http://wiki.audacityteam.org/wiki/About_Audacity#Short_history

http://www.audacityteam.org/about/features/

http://manual.audacityteam.org/o/index.html

https://en.wikipedia.org/wiki/Audacity_(audio_editor)

http://www.afterdawn.com/software/version_history.cfm/audacity

http://manual.audacityteam.org/o/man/mixing.html

https://docs.coronalabs.com/daily/guide/start/introLua/index.html#memory-allocation

https://en.wikipedia.org/wiki/Lua_(programming_language)

http://www.lua.org/uses.html

Software Design & Development 2016
AS1 - Hardware / Software Presentation

Ryan Oo, Jai Raina, Andrew Wong

Sound Processing - Summary

Overview

- Method of capturing sound needs to be made for computers as they use binary
- Microphones do this
- Two main types of microphones are Dynamic and Condenser microphones

Condenser Microphones

- Have diaphragm (also known as front plate) and a stationary back plate
- Diaphragm moves in and out in response to sound wave
- Back plate remains still as current flows between the diaphragm and the back plate
- Electrical current alterations reflection soundwave. Current sent to an ADC (analogue to digital converter) which transfers the electrical current into digital data

Dynamic Microphone

- Three main components; diaphragm, copper coil of wire and magnet
- Diaphragm attached to copper coil of wire, which is surrounded by a magnet
- Diaphragm moves in and out in response to the soundwave and so does the copper coil, creating an electric current with the magnetic field from the magnet
- Electric current sent to an ADC to transform into digital data

Sound Sampling

- Electric current from microphones fed into capacitor, ADC measures current in capacitor
- SAR (successive approximation register) produces binary number eg. 225 to 0
- DAC (digital to analogue converter) transforms binary number in to electrical current
- Comparator compares electrical current in DAC to electrical current in capacitor
- If same electrical current, comparator informs SAR to store number generated
- Typical sampling rates; 22,000Hz and 44,100Hz

Reproducing Sound

- Amplitude of sound samples changed into electrical current
- Speaker contains diaphragm connected to copper coil of wire surrounded by magnet
- Electrical current causes copper coil to vibrate in response to fluctuating magnetic field
- Causes diaphragm to move, creating compressions and rarefactions in the air we hear as sound

Historical Development

- **Acoustic Era:** Phonograph for recording sound. Poor quality. Frequencies of 250Hz to 2,500Hz recorded
- **Electronic era:** Electronic microphones, amplifiers and electronic disc cutters. Better quality sound. Frequencies of 60Hz to 6,000Hz recorded
- **Magnetic Era:** Magnetic tape recorder. Improved sound fidelity, more portable
- **Digital era:** Binary, sound sampled at 44,100Hz. USBs, Hard Drives and CDs used to store sound. High quality sound recorded. Can capture all frequencies humans can hear

Ryan Oo, **Jai Raina**, Andrew Wong

Sound Editing - Summary

Brief Overview of Sound Editing

- Sound editing has been present in all of society for decades
- Movies, television, games and music require sound editing
- Achieved through 2 key ways: analogue editing and digital editing
- Examples of digital sound editing software include: Audacity, Wavepad, FL Studio, Adobe Audition
- Presentation will be focused on Audacity

How Audacity Works

Audacity is a multi-track audio editor which helps users to create and edit audio files

- In the diagram shown on in the presentation, these are what the numbers represent:
    1. Menu Bar
    2. Transport Toolbar
    3. Tools Toolbar
    4. Recording Meter Toolbar
    5. Playback Meter Toolbar
    6. Mixer Toolbar
    7. Edit Toolbar
    8. Transcription Toolbar
    9. Device Toolbar
    10. Timeline
    11. Track Control Panel
    12. Audio Track
    13. Label Track
    14. Selection Toolbar
    15. Status Bar

Editing Features of Audacity

- Simple editing using the cut, copy and paste buttons
- Unlimited sequential undo and redo
- Mixing and compiling as many tracks as users need
- Use of sound effects

Historical Development of Audacity

- Audacity was first worked on by Dominic Mazzoni and Roger Dannenberg in 1999, as a research project for university
- First released on the 28th of May, 2000 as version 0.8
- Near the end of 2003, Audacity released the 1.2 series, which introduced hundreds of new features
- After the run of the 1.2 series a Beta 1.3 series was released which served as a foundation for the currently running 2.0 series
- By version 2.0, Audacity was highly enhanced and improved from its predecessors as it is now smoother, more effective and has more variety for users to work with

Ryan Oo, Jai Raina, **Andrew Wong**

Lua - Summary

Overview

- Cross platform, extension program language – extends the functionality of existing software
- Developed by Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes
- Programming language is similar to other languages
- 'Lua', opposite of the Portuguese word 'Sol' (sun)
- Won the 2011 Game Developer Magazine Front Line Award

Programming Syntax

- Formatting is not strict (*ahem* Python)
- Semicolons optional
- Relatively common syntax
- 1-based arrays (arrays start at 1)
- No difference between integers and floats
- Undefined values become nil.

Variables

- Variables aka identifiers store data
- Global variables – publically accessible
- Local variables – accessibly only in its nest
- Naming – Alphanumeric + Underscore; Cannot start with a digit; Cannot be a Lua keyword

Memory Management

- Memory leak -> less available ram -> crash?
- Garbage Collection – Automatic memory management

Advantages

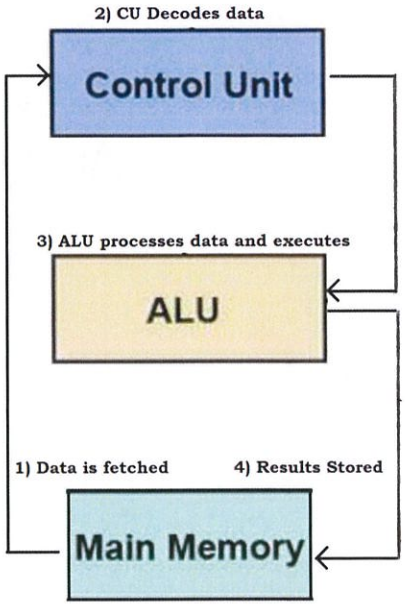
- Prototyping
- Shorter development time
- Compiled at runtime
- Lightweight (overhead) (*cough* Java)
- Relatively simple and easy to read syntax (pseudocode)

Applications

- VLC Player
- TeamSpeak 3
- Garry's Mod
- CryEngine 2
- Adobe Photoshop Lightroom
- World of Warcraft
- LEGO Mindstorms NXT

CPU

Central Processing Unit



There are two main parts are located in the CPU, the ALU (Arithmetic Logic Unit) and CU (Control Unit). The ALU can be referred to as the "brain" of the CPU, calculating the information, interpreting them, converting them and responding using a series of algorithmic and complex instructions to execute programs, while the CU is the "control center", giving commands to the "brain" and translating the program data into readable instructions for the ALU

Fetch – Decode – Execute – Writeback

Fetching is retrieving the instructions from a program, the instructions are very complex, and over a million sets are given.

Decoding is translated by the CU, the information is presented in a series of computer commands.

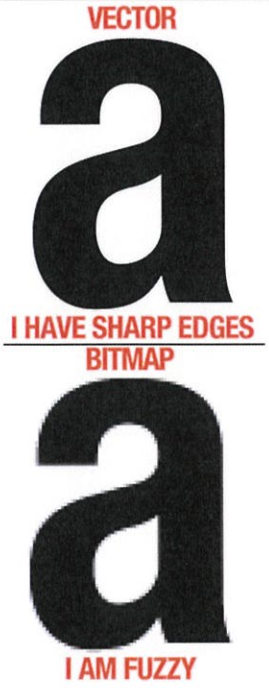
Executing the instructions is done by the ALU, resulting in a different set of information compared to the original fetched information, this information is then-

Stored back into memory, the writeback

Graphics Software

Photo manipulator and; or generator

Split into two main groups for specific reasons



Bitmap – To manipulate images

Bitmap is the one to go to if you need a quick edit on an image that maybe seems a bit too small or maybe the colour seems off. It is mainly the most used out of the two and even most of the major editing software include vector on a basic level.

Vector – An assembly of polygons to generate an image

This would be the software used to generate detailed, scaled images that have no limit to their detail, such an application is capable of generating images with maximum resolution both zoomed in and zoomed out.

HTML

HyperText Markup Language

One of the most basic variations of computer programming language. HTML is the basic structure of most websites, having the potential to be complex and elegant based on the amount of time and sophistication put into it. HTML can collaborate with other programs (JavaScript) to obtain effects such as animation. HTML uses a very simple coding structure and can be generated with basic text based programs such as Notepad.
This .html file can then be opened with internet browsers to interpret this information and form a website based on it.

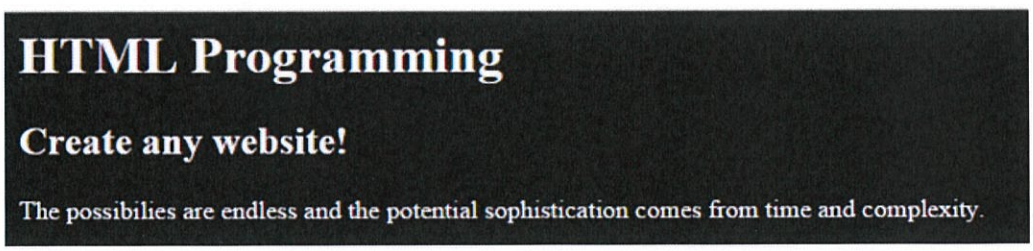
```
<html>
<title>HTML Programming</title>
</head>

<body bgcolor="#000000">
<font color="#FFFFFF">

<h1>HTML Programming</h1>
<h2>Create any website!</h2>

<p>The possibilies are endless and the potential
sophistication comes from time and complexity.</p>

</body>
</html>
```



Jackie Mock

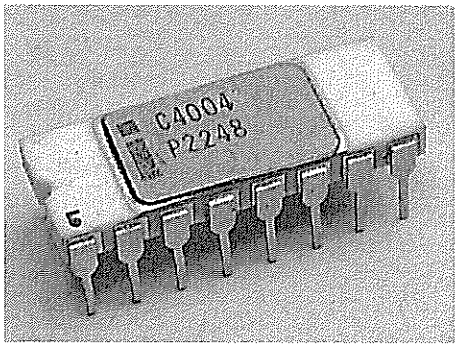
CENTRAL PROCESSING UNIT (CPU)

WHAT IS THE CPU?

The Central Processing Unit (CPU) or also commonly known as microprocessor is the core, or the brain of your computer. It processes instructions that it receives from decoding the code in programs and other such files, and then sends back the computed results to carry out the given instructions. Thus we can determine that the CPU has four main primary functions: **fetch, decode, execute,** and **writeback.**

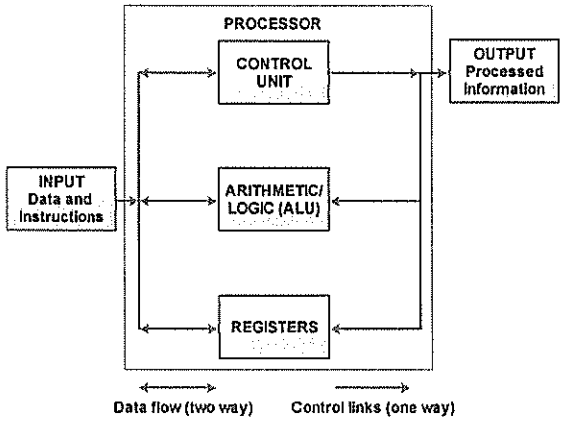
BREIF HISTORY OF THE CPU

The CPU was first invented by world famous company **Intel** alongside **Ted Hoff** and others in the early 1970's. The very first CPU released by Intel was called the "**4004 Processor.**" It was a 4-bit, 16 pin microprocessor that operated at a speed of 740KHz and at roughly eight clock cycles per instruction (explained in the near future), meaning that it was able to execute up to 92,600 instructions per second. But when compared to your everyday CPU that runs at for example, 2.8GHz, which is 2,800,000,000 cycles per second, technology really has come a long way.



COMPONENTS OF THE CPU

The CPU contains two main components. These are the Arithmetic Logic Unit (ALU) and the Control Unit (CU). The ALU performs mathematical, logical and decision operations whilst the CU directs all of the processor's operations.



- **Intel 4004** consisted of 2,300 transistors and performed 60,000 operations per second.
- An average **Intel Pentium Processor** of today, consists of 3,300,000 transistors and performs approximately 188,000,000 operations per second.

FOUR PRIMARY FUNCTIONS OF THE CPU

FETCH

In this step, the CPU retrieves the instructions from program memory that it needs to run. Every instruction in a program is stored at a specific location (address). The instructions address in the program memory is determined by a **Program Counter (PC)** which assigns a number that labels the address of the next instruction to be fetched or accessed.

DECODE

A CPU will act according to what it is told to do, therefore what instruction the CPU fetches from program memory determines what the CPU will. Programs can be written in various forms or types of code, however the **compiler** for that specific language breaks the code down to **Assembly Language**. **Assembly Language** may vary between differing CPU's. Next, an **"assembler"** converts Assembly Language into binary code, which is what the CPU can read in order to perform the given instructions.

EXECUTE

The execute step is where the CPU performs the instructions it was given after **fetching** the instructions and **decoding** them. Depending on the instructions that it was given, the CPU can do multiple things:

- Calculate extremely complex mathematic functions using the ALU
- Move data from one memory location to another
- Jump to different addresses in the program based on decision made the CPU

During each action, various parts of the CPU are electrically connected so that it can execute all or part of the instructions it was given, until it is completed.

WRITEBACK

Generally when the CPU takes actions, it creates some **"output"**. The CPU takes this **output** and writes it in the computer's memory. After executing an instruction, the **output** would be written back into a specific address. However, the **program counter** would change to reflect the start of the next set of instructions.

Christopher Rice.

HTML RESEARCH

HTML (HyperText Markup Language) is the computer language that allows for website creation and website editing. HTML is essential for all websites as it is the skeleton of the website, containing the structure, layout and formatting of the webpage as well as any links present on the website. HTML is one of the easiest computing languages to learn as the steps to create a simple website through HTML are relatively basic and easy to understand, while also being powerful enough to be able to produce complicated webpages that look professionally produced.

HTML is often paired up with CSS (Cascading Style Sheets) and JavaScript in order to add specific formatting to a website or interactions with certain aspects of the webpage. More specifically, CSS controls how the HTML pages are presented or how they appear, and JavaScript allows for special effects for websites (mainly used in professional productions). HTML is still the most crucial part to a website as everything from the text input, headings, images, links (through hypertext) and other elements of a website. Changing the HTML coding of a website will affect how the entire webpage will appear and function, as well as altering any coding input by means of CSS or JavaScript.

HTML is defined as Hypertext Markup Language, and is split into 3 separate terms to form the defining features of HTML; the *hypertext* which is the main method of moving around on the web, from clicking on hyperlinks that bring you to the next page or anywhere else on the internet it is linked to; the *markup* which is what the HTML code does to any text within a "HTML tag" for example bolding or italicizing text; and the *language* which is what HTML is defined as, a computing language that contains specific code-words and syntax that allows programmable functions to create and edit like any other computing language.

HTML consists of a series of short codes typed into a text-file by the site author. These are called the "tags". The text is then saved as an .html file, and upon completion can be viewed through a browser, like *Internet Explorer* or *Mozilla Firefox*. This browser then reads the file and translates the text into a visible form, if successful then loading and displaying the page as the author had intended. HTML pages can be created or edited with something as simple as a text-editor such as Notepad++ or powerful graphical editing software to create these HTML files and upload them as websites.

The 'tags' that exist within a HTML file change the entire layout of a line, section, or the whole page depending on how and what is put within these tags. A HTML tag is defined as text or values within two sharp brackets that corresponds to something within the file ("<...> OR </...>"). Tags have different properties, and are classed as either container elements, or empty elements. The main difference between these two is the inclusion or lack of a "start" or "end" tag bracket. Empty elements do not contain an end bracket, as they usually do not have content within the tag to actually close it. For example the line break tag
 is just a command placed within lines of coding to ensure that the text following is placed onto the next line, and does not correspond to anything other than the command to start the next line immediately, hence why it does not need to have an end tag of </br>. Both of these tags can be included within the same line of coding, and can be classed as a nested element; where one tag is "nested" within another larger tag. For example a ... tag that starts a paragraph could have a ... within it to make a specific part of the paragraph appear bold.

Any file created with HTML begins with the tag "<!DOCTYPE...>" and declares the current version of HTML the file was created by to the web browser rendering the file. The most current and widely used version of this language is HTML5 and the declaration tag is viewed as "<!DOCTYPE html>".

Immediately after this declaration tag, the "<html>" tag is placed in the line just under it to indicate the beginning of the HTML document, and "<html/>" indicates where the file terminates, closing up the HTML coding. The <head> and the <body> tags follow the starting <html> tag, and indicate the two main parts of the HTML file. The "head", which includes the Title and anything else that would be included in the header of a regular document. None of the formatting or text that appears on the interactive webpage would be included in the <head>; instead it would be placed in the <body>. The "body" is where all the images, text, hyperlinks and other features and information of the webpage can be seen. The <head> and <body> are the largest tags that appear within the document. Most of the other tags are there just for formatting and keeping the page neat and tidy.

Structural Tags

Start Tag : End Tag	Content
<!DOCTYPE>	- Defines document type and HTML version
<html> ,</html>	- Encloses complete HTML document, containing the **<head>** and **<body>**
<head>,</head>	- Represents the header of the document, containing tags such as <title> and <link>
<title>,</title>	- Title of document, what the browser displays in the tab
<body>,</body>	- Represents the "body" of the document containing tags like <h1>, ,
,	- Contains a heading, size of the number indicates how large/small text is (1=largest)
,	- Contains paragraph
,	- Used to group sections of HTML (multiple lines of coding)
,	- Used to group sections of HTML (inline only)
<!--*insert*-->	- Used for comments within the HTML document, will not appear on webpage

Formatting Tags/Attributes

Start Tag : End Tag	Content
	- Allows for specifications on formatting as in CSS eg. <"font-family; arial; color:#ABCDEF;"> etc.
,	- Makes text bold
,	- Makes text italicised
<u>,</u>	- Underlines text
<strike>,</strike>	- Makes a "strikethrough" in the text
[,]	- Makes text superscript
_,	- Makes text subscript

Andrew Castillo, Wisley Chau, Jason Cheng

Seminar Notes

Graphics Processing

- A GPU is specialized for compute heavy tasks, such as video rendering.
- All frames drawn by the GPU are rendered into a frame buffer of video random access memory (VRAM) which stores the frames until they are ready to be displayed. Due to this system of rendering more VRAM increases the performance for higher resolutions being rendered.
- GPUs produce much more heat than CPU's when used for heavy workloads like rendering a game. Thermal throttling by reducing performance and clock speeds when under high temperatures will occur to maintain safe temperatures unless a cooler can be attached to the unit itself.
- Outputs include HDMI, RGB/VGA, DVI and DisplayPort. The outputs are generally located at the end of the card where the card can be screwed into the case.
- Graphics cards use the PCI-Express slot located under the CPU on the motherboard.
- They are much more effective than CPU's for processing large visual workloads which are done in parallel
- GPU-accelerated programs allow the CPU and GPU to work synchronously by offloading compute workloads onto the GPU
- GPU's have several thousand CUDA cores compared to relatively low amount of cores within CPU's (mostly between 1-8 for consumer grade chips)
- The large amount of cores makes the GPU much better for parallel tasks.
- Several GPU's can be utilized in the same system through either NVidia's scalable link interface (SLI) or AMD's Crossfire technology.
- This allows for higher performance however compatibility is fairly limited and performance gains are not simply added but only a percentage of performance can be achieved.
- The more cards added to the array, the lower the performance scales, however the higher the resolution of the task being rendered, the better the scaling.
- Integrated graphics processing units are the most popular currently, they are usually packaged with most of Intel's Ironlake, Sandy Bridge, Ivy Bridge, Haswell, Broadwell and Skylake processors, and within AMD's accelerated processing units (APU)

- IGPU's are not their own card themselves but are inside the CPU and can be used by outputs in the motherboards I/O.
- Discrete graphics card solutions are plugged into the PCI-E card slots which have bus speeds of 4x, 8x and 16x.
- The main components of a GPU are the GPU itself, VRAM, MOSFETS,
- They are used mainly for better performance than other types as they can have heatsinks, direct liquid cooling onto the GPU, cooling for MOSFETs and VRAM and generally much more efficient heat dissipation systems than on integrated units

Game Engines Summary

What is a Game Engine?

Game engines combine numerous types of software to present a video game. Many different game engines exist with different strengths and weaknesses, such as a focus on 2D and 3D. Most game engines are made in-house by developers for use in their own games. Most game engines require the use of other programs to produce content for them, such as sound and textures.

Components of a Game Engine

Game engines have 4 main components with distinct roles.

- Renderer – draws all visuals using textures, 3D models and shaders (rendering code). Uses OpenGL (cross-platform) or DirectX (Windows and Xbox).
- Audio – processes audio effects such as reverberation. Sometimes also can be used to blend music dynamically. Often uses FMOD (dynamic audio) or WWise (HDR audio).
- Physics – uses mathematical algorithms to simulate movement of objects, such as gravity, cloth movement, water simulation and destructible objects. Uses Havok or PhysX (can use NVIDIA GPUs to process instead of the CPU).
- Game Code – written per-game by developers in a large variety of languages. Handles all other elements of the game, such as weapons and enemies. Also handles loading files and saving the game.

Game Engines Used Today

- Unreal Engine (UE4) – developed by Epic Games
  - Released in 2012, public release in 2014, free in 2015. Latest version (4.10) released 2016.
  - Used on hundreds of games, such as Borderlands and Rocket League
  - Supports all operating systems, game consoles, iOS and Android.
- Unity – developed by Unity Technologies
  - Released in 2005; current version (5.3) released in early 2016.
  - Free limited version and fully-featured Pro version.
  - Used on thousands of games such as Hearthstone
  - Runs on the same platforms as Unreal Engine.
- Source Engine – developed by Valve Corporation
  - Released in 2004 and updated periodically over time.
  - Used on all post-2004 Valve games as well as some third-party titles i.e. Insurgency and Titanfall.
  - Free to use for modifications & free games, full licence for large developers.
  - Supports last-generation game consoles, all operating systems and Xbox One.

Graphics Processing Summary

- A GPU is specialized for compute heavy tasks, such as video rendering.
- Started with NVidia which introduced the GeForce 256.
- Several manufacturers have come and gone since, now with only AMD and NVidia remaining.
- Past manufacturers include 3DFX (now bankrupt), ATi (now AMD), Broadcom, ARM Holdings, and more.
- A GPU renders frames into a frame buffer so volatile performance tasks can still be rendered effectively.
- GPUs produce much more heat than CPU's when used for heavy workloads like rendering a game. Thus, a cooler must be used for more efficient performance.
- Display outputs include HDMI, RGB/VGA, DVI and DisplayPort.
- Graphics cards use the PCI-Express slot which is most efficient at sending information with low latency to the CPU.
- They are much more effective than CPU's for processing large visual workloads which are done in parallel
- GPU-accelerated programs such as Adobe After Effects, Sony Vegas and Photoshop all allow the CPU and GPU to work synchronously by offloading compute workloads onto the GPU
- GPU's have several thousand CUDA cores making it much better for parallel tasks.
- SLI or Crossfire technology now unlocks the possibility of using several video cards.
- The more cards added to the array, the lower the performance scales, however the higher the resolution of the task being rendered, the better the scaling.
- Integrated graphics processing units are the most popular currently with over half the market share.
- IGPU's are not their own card themselves but are inside the CPU and can be used by outputs in the motherboards I/O.
- Discrete graphics card solutions are plugged into the PCI-E card slots which have bus speeds of 4x, 8x and 16x.
- The main components of a GPU are the GPU itself, VRAM, MOSFETS,
- They are used mainly for better performance than other types as they can have heatsinks, direct liquid cooling onto the GPU, cooling for MOSFETs and VRAM and generally much more efficient heat dissipation systems than on integrated units

Game Engines

- Game engines are extremely complex combinations of numerous programming and artistic disciplines, which together form the presentation of a video game.
- Different types of game engines exist with specialised roles and features.
- Hundreds of game engines exist with different strengths.
- Some developers make engines for their own games; others make engines as products to sell.
- Game engine software alone is not fully capable of making a game – programs such as Photoshop are often used to create visual assets, and other programs are used to create sound, 3D models etc.
- Game engines can be distilled to four core components, which are largely universal:
    - The renderer is tasked with utilising a rendering API (OpenGL or DirectX) to interface with the GPU to create visuals from 3D models, textures (2D images) and shaders (rendering code written in specialised programming languages, such as HLSL or GLSL). Shaders are used to add special effects to the image, such as smoothing edges or shading.
    - The audio component is responsible for processing all sound and music in a game. It can add effects like echo, Doppler and dynamically blend between sounds to respond to player input. This is often used for music. FMOD and WWise are two often-used third-party audio packages used to handle audio in game engines.
    - The physics component handles the simulation of all physically-moving objects in the game such as cloth and water. Middleware often used includes Havok and NVIDIA PhysX, which can run on their GPUs.
    - Game code runs the rest of the game, controlling logic and all objects seen in the game. Use many different programming languages depending on the engine. They also often feature high-level scripting for easy and efficient game creation.
- Three well-known game engines available for public use:
    - Unreal Engine 4 – developed by Epic Games, released in 2012, publicly released in 2014 and made free in 2015. Previous Unreal Engines were used on games such as Borderlands and Rocket League. Unreal Engine 4 runs on all major gaming platforms.
        - Game view – shows view of game world to developer.
        - Content browser – manages all game assets.
    - Unity – developed by Unity Technologies, released in 2005, currently at version 5.3. Features a limited free version and a fully-featured Pro version for $75/month. Used on game such as Hearthstone and Fallout Shelter.
        - Game view – shows game world

    - Much more simple UI than Unreal
- Source – developed by Valve, released in 2004, new versions released periodically since then. Used on all recent Valve games (i.e. Counter-Strike Global Offensive) and some third-party titles. Free version for mods and expensive paid version for commercial games.
    - No all-in-one development tool – uses many smaller programs and plugins.
    - Main program – Hammer editor for level editing.

C++ Programming Language Summary

Development of the C Programming Language

The C Language, developed in 1972 by Dennis Ritchie at Bell Telephone Laboratories, was built to be a high-level language that allowed access to memory and to create efficient code. In 1978, Dennis published 'The C Programming Language', making the standard for C. Other versions that revised the language like C89 and C90 revamped the language, with the survivability of the language a testament to its quality.

Development of the C ++ Programming Language

The C++ programming language as developed in 1983 by Bjarne Stroustrup, also in Bell Telephone Laboratories. C++ improves on many of the base features that C offers, as well as providing more in terms of content. It was ratified in 1998 by the ISO committee and was majorly revised in 2014, with the version called 'C++14', the current present day version in use.

Fundamentals of C++

C++ allows for three programming paradigms:

- Object Oriented - Defining objects that contain data, allowing classes. Classes are 'blueprints' of objects, allowing code to be more compact and organised.
- Functional – Treating the code as the evaluation of mathematical functions.
- Procedural – Concept of procedural calls, allowing for multiple modules.

Examples of C++ programs Used Today

- Microsoft Office
- Adobe Photoshop
- Indesign
- Firefox
- Google Chrome
- Python
- Perl
- Facebook (Major backend written in C++)
- Amazon (Major backend written in C++)

The Tiobe Index

The Tiobe index is a yearly measurement on which languages are popular in the global community. C++ has moved up from place 4 to 2 in the March 2016 index.

C++

- The C language is the precursor to the C++ language, introduced as a language that advertised access to memory and efficient code management.
- C++ was released after C, adding new features to the C language as a standalone language. It has survived multiple major revisions.
- C++ allows programming in multiple different programming paradigms:
    - Object Oriented
        - Introducing 'objects' and 'classes', allowing code to be more compact and organised.
    - Functional
        - Treating the code as an evaluation of mathematical functions.
    - Procedural
        - Procedure calls, and structuring solutions by modules.
- C++ is still a relevant language and is still moving up the TIOBE index.

David Wu / Kerry Wu Pg1

Summary

The **Raspberry Pi**'s concept came to existence during the 1990s. When a few people working at the University of Cambridge, Eben Upton, Rob Millings, Jack Lang and Alan Mycroft, realised that student aren't applying for Computer Science due to the fact that computers were too expensive for the sole purpose of programming. They worked on the idea since, and concentrated in modifying another version of Linux called Capsian at around the 2006-2008, but the first Raspberry Pi model wasn't released until 2012. There are current 4 models of Raspberry Pi, ranging from $5-35. Although it's not powerful enough to do any resource intensive work, it can act like miniature computers, having CPU that can handle 900MHz, a GPU and multiple ports that connects to many media. The Raspberry Pi has since been used for many purposes including modifying an X-Box, taking photos above the clouds and more.

3D modelling and animation are advancement in technology that will affect our lives, especially in the entertainment industry and in industrial settings. It's responsible for almost all games ranging from the AAA to the low budget and many good 3D animated. It is also used to creating models to aid designers and allows both the designer and customer to gain an better understanding of that the final product will look like (or feel like, if the model is used for **3D printing**) and how it contrast will the idea that they had in mind. The two most famous software for 3D modelling and animation are **Autodesk Maya** and **Blender**. Maya is a professional software that sets the industry standards having achieved 3 awards since its inception. Blender is a free professional software that is popular among the indie developers and artists, but it's also used by NAYA to created publicly available models. There are also other 3D creation tools out there, such as **ZBrush**, **Creo** and **3DS Max**, but they are generally outshined by Maya and Blender's glory.

C# is developed by Microsoft and had its first appearance about 16 years ago in the year 2000. C# has many standards, including being a simple, modern, general purpose object-oriented language. Which means that it's suited for beginners and allows one to develop simple apps very easily, on a **Windows** computer, as it does not allow cross-platform development just yet. Although Microsoft is making it's steps and it took over Xamarin, allowing for mobile development using C#. It's a compiled language, allowing it to benefit from having a faster run-time than most language with the exception from C, the fastest language around. C# is also a multi-paradigm language, which allows the programmer to work freely in a variety of style and decide on what suit them best. It's created after image of C++ and Java, with the intention to take the best feature which it certainly does, with many features that are not included in the original two languages. C#'s benefits outweigh its problems, especially if you're developing on a Windows.

David / Kerry Pg1

Seminar Notes

Raspberry Pi:

- The design goal for the Raspberry Pi was to create a cheap, yet functional processor which allows for programming
- The concept was created in the 1990s, by people working at the University of Cambridge, Eben Upton, Rob Millings, Jack Lang and Alan Mycroft, when they discovered that there was a significant decline of student applying for Computer Science due to computers being too expensive for the sole purpose of programming
- It utilise a modified version of Linux created by the team called the Caspian that was developed from 2006-2008
- There are currently 4 different models of Raspberry Pi existing on the market, ranging from $5-35, averaging at $20 with around 512mB of RAM, a CPU that is capable of handling 900MHz, a GPU, USB 2.0 Ports, video input, video output, audio input, audio output, 'on-board' storage, 'on-board' network, and about 30 grams.
- Raspberry Pi cannot host servers due to its 512mB of RAM, and thus it also cannot perform any resource consuming task
- Raspberry Pi can be connected together, allowing it to do more than what it's originally intended
- People have had many ingenious uses with the Raspberry Pi as it is cheap and efficient:
    - Someone mounted an X-Box controller on the Raspberry Pi Zero, allowing him to play retro games in HD with a portable X-Box controller
    - A balloon enthusiast, Dave Ackerman send his Pi to the edges of Earth's atmosphere, using a weather balloon. It endured temperatures of -50 degrees Celsius and 1% oxygen. He has even gotten it take pictures while it was in flight as shown
    - Evan Cohen bought a one-way mirror and removed the back. He then, added a programmed Raspberry Pi with his software and hooked it up to a power source, allowing voice control with the mirror, which is connected to his house

David / Kerry Wu Pg 2.

3D Modelling and Animation:

- 3D modelling "is the process of developing a mathematical representation of any three-dimensional surface of an object via specialized software"
- Many 3D modelling software also allows creating animation with in application
- 3D modelling software are used to create models for rendering, creating animations, importing as game assets and for 3D printing
    - Artistic:
        - 3D modelling and animation software are responsible for the transition from 2D to 3D animation
    - Gaming:
        - Most high-budget games are 3D games, using models made from 3D modelling software to import as assets
    - Industrial:
        - 3D models are created for industrial design to help the designers better understand and design their project
        - The models can be printed to give a more authentic feeling and allows the designer and customer to see mistake they otherwise wouldn't have in plain drawing
        - The models can also use used for promotional purposes, and could be animated for better effects
- There are many 3D modelling and animation software, two of them are most well-known:
    - Maya:
        - Developed by Autodesk Inc.
        - Written in C++, MEL, Python and C#
        - Initial release in 1998
        - Industrial standard
        - Involved in the development of many high budget films (The Matrix, South Park, Game of Thrones, various Disney and Pixar films and more)
        - Gained 3 awards since its inception on 2003, 2005 and 2008 regarding motion picture and technical achievements
    - Blender:
        - Developed by the Blender Foundation
        - Written in C, C++ and Python
        - Initial release in 1995

David / Kerry Wu Pg3

        - Free, open source software
        - Used by NASA for developing public models
        - Has internal game engine
        - Allows coding in python
    - Other 3D modelling and animation software include:
        - MAXON's Cinema 4D (known for creating 3d animations)
        - Pixologic's ZBrush (known for high-detail sculpting and human creation)
        - PTC's Creo (used for detailed industrial models, creating allowing 3D printing)

- To tackle this issue, Microsoft has brought Xamarin studios, which specializes in mobile development using C#, allowing C# to develop mobile applications, this is a promising step towards cross-platform development

Programming Language: C#

- C# was developed by Microsoft, with a team form by Anders Hejlsberg
- Its first appearance was 16 years ago in the year 200 and is known as C# 1.0. The current version, C# 6.0's stable release was on July 20th, 2015
- Its original name was "COOL", which stood for "C-like Object Oriented Language", but was changed due to fear of copyright
- ECMA has many standard design goals for C#, including:
    - C# is intended to be a simple, modern, general-purpose, object-orientated language.
    - The languages, and its implementations, should provide support for software engineering principle such as array bound checking, detection of attempts to use uninitialized variables and automated garbage collection.
    - The language is intended for use in developing software components suitable for deployment in distributed environments.
    - Source code portability.
    - Support for internationalization.
    - C# is intended to be suitable for writing applications for both hosted and embedded systems.
    - C# was not intended to compete directly on performance and size with C or assembly language.
- C# is a compiled language, allowing it being able to develop faster application at run-time
- C# is a multi-paradigm language, allowing the programming to work freely in a variety of different styles and that they choose whatever tool is best for the job and best suited for themselves.
- C# was created after the image of C++ and Java, with the intention to take only the good bit from both languages.
    - Comparing to Java, it is much less verbose, it also has unsigned types, operator overloading, delegate and more
    - Comparing to C++, it does not require the developer to worry about memory management, it doesn't require header files
    - C# has .NET framework, which is not available to both languages preceding it, along with the inclusion of formalized concept of get-set methods, making the code more legible
- C#'s biggest problem would be that it's not cross-platform, and it is only available for developing windows executables