

# Wave File Discrete Fourier Transform

Yong Wen Chua (ywc110)

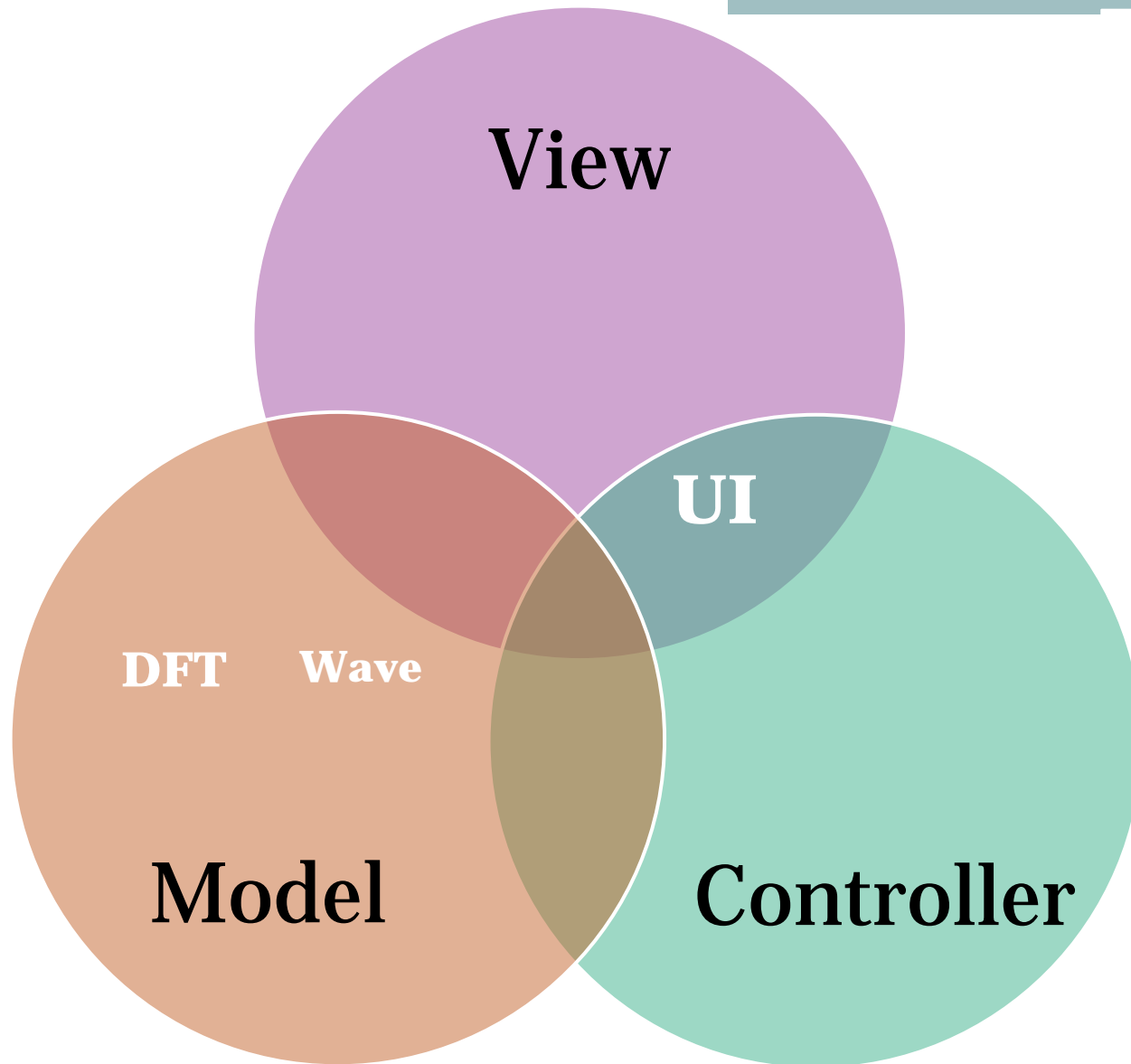
A series of horizontal lines of varying lengths and colors (teal, light blue, and white) extending from the left edge of the slide towards the right, positioned below the author's name.

# Features

- Analyse Wave File
- Do Discrete Fourier Transform of Wave File  
sound wave using the Fast Fourier Transform  
Algorithm in Matlab
- Ability to edit Wave File data directly and save  
the file

# Design

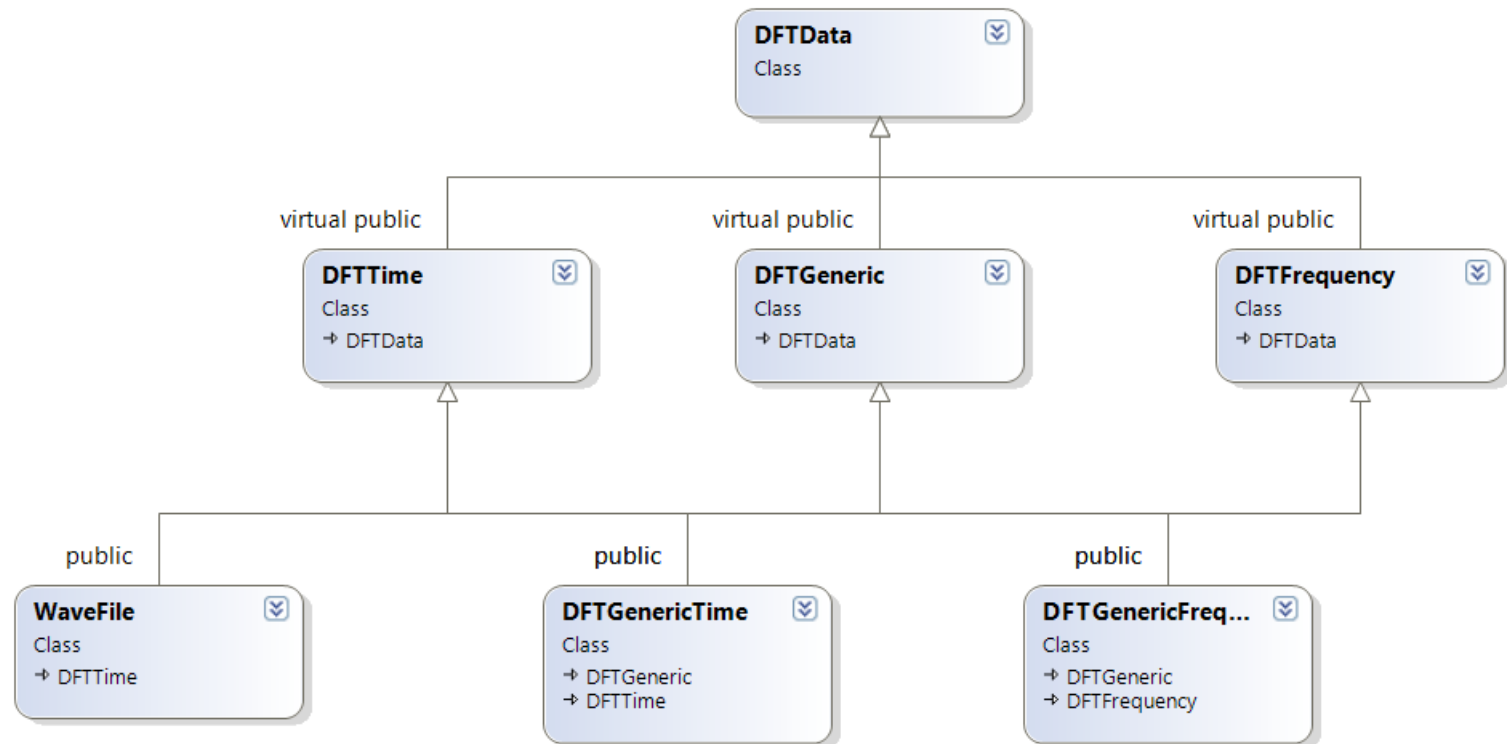
- **Three Separate Modules**
  - Discrete Fourier Transform (DFT)
  - Wave File (Wave)
  - Console Interface (UI)
- **Loosely based on Model View Controller**



# DFT Namespace

- Defines Abstract Data Types and Abstract Base Classes
- Purpose:
  - Define a set of generic data types (in time or frequency domains) for classes to use
  - Define a set of generic DFT classes for others to use
- Allow for consistency across different DFT algorithms or data storage

# DFT Namespace



# DFT Namespace

- **DFT::DFTData**
  - An abstract base class.
  - Defines a set of publicly exposed methods for all DFT methods to use to retrieve and store data
- **DFT::DFTTime & DFT::DFTFrequency**
  - Derived abstract class
  - Used to signify that an object has time domain or frequency domain data respectively
  - Wave::WaveFile derives from DFTTime

# DFT Namespace

- **DFT::DFTGeneric**
  - An abstract derived class that implements methods described in DFT::DFTData
  - A sort of generic and “default” container that programmers can use in lieu of coding a new one
- **DFT::DFTGenericTime & DFT::DFTGenericFrequency**
  - Derived class from DFT::DFTGeneric to signify storage of time or frequency domain data respectively using the DFTGeneric implementation



# DFT Namespace

- DFT Base Class for different algorithms of DFT
- Only one “algorithm” – Matlab FFT – is implemented
- Inefficient “brute force” algorithm was too slow and abandoned.
  - But it was derived from the DFT base class



# Wave Namespace

- **Wave::WaveFile**
  - Inherits DFT::DFTTime
    - Essentially means that WaveFile can be used in DFT classes to represent an abstract data type consisting of time domain data
  - Is a giant class that acts as an Abstract Data Type for a Wave File
  - Includes parsing and handling of the Data type

# Wave Namespace

- **Wave::WaveChunk**
  - An abstract data type for a “chunk” in a Wave File.
  - Manipulation methods
  - Used by Wave::WaveFile for internal representation and storage of data
  - A template class:
    - Allows the “chunk” to store its data in any form.
    - Defaults to `vector<Wave::Word>`

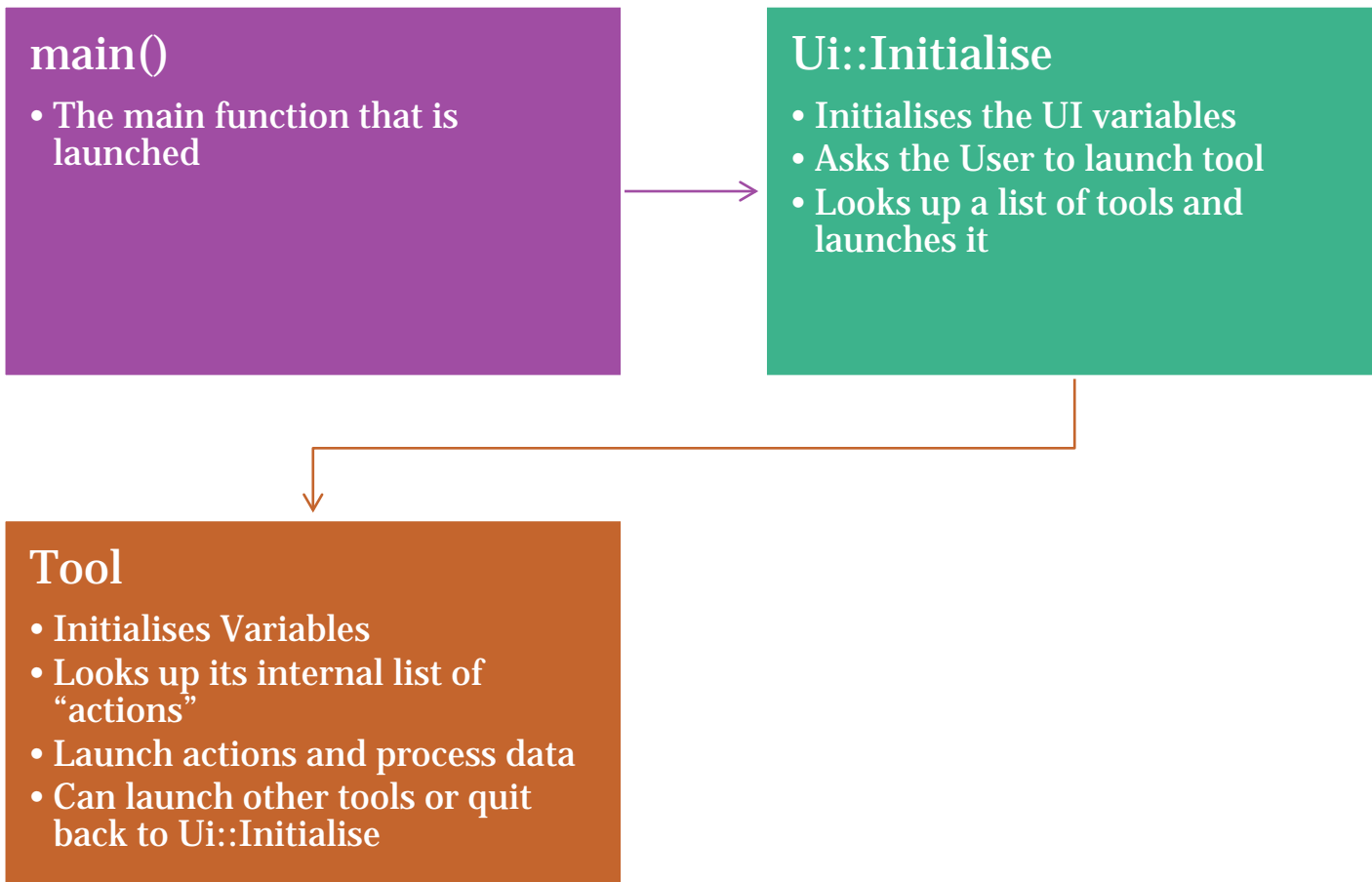
# Wave Namespace

- **Wave::Word**
  - An abstract data type for a “word”, equivalent to an array of four characters → 4 Bytes.
  - Manipulation methods
  - Used by Wave::WaveFile for internal representation and storage of data

# UI Namespace

- Provides a “console interface” to the user
- Instantiates and initialise necessary data types and objects to perform operations
- Modular approach

# UI Flow



# Ui Modules Basic Design

- Consist of a “main method”
- Stores a vectors of “sub-modules”
- Asks for user input and launches the “sub-modules”
- Sub-modules can launch other “modules”
- Possible to cascade several “modules in modules”

# Screenshot Examples

```
H:\Documents\Year 1\Computer Labs\Spring Term\Project\Src\x64\Debug\WaveDFT.exe
Welcome to the Wave Tool.
The following commands are available.
Type 'help command' for help on a particular command
    copy    dump    help    info
    load    matlab  open    quit
    unload  write
-----
wave> open files\eg1.wav
File opened and parsed. Use 'info' to display information about the file.
wave> info
Wave File Details:
- File open and associated
- Stream data not loaded
- Data Stream Size: 93972 Bytes
- Channel Size: 2
- Sampling Rate: 8000 Hz
- Sampling Interval: 0.000125 s
- Block Size 4 Bytes
- Sample Size: 16 Bits
- Sample Count: 46986
- Block Count: 23493
Frequency Domain Details:
- Not Loaded
wave> load
wave>
```

```
H:\Documents\Year 1\Computer Labs\Spring Term\Project\Src\x64\Debug\WaveDFT.exe
-----
Welcome to the Wave Tool.
The following commands are available.
Type 'help command' for help on a particular command
    copy    dump    help    info
    load    matlab  open    quit
    unload  write
-----
wave> matlab> wave> matlab> wave> matlab
WARNING: Any change you make in the Matlab module will cause a change in this module. You also cannot change the number of channels (dimensions) or blocks (interval number) in Matlab. If you want to create a new file from this wave as a basis, use copy.
Press ENTER to continue...
-----
Welcome to the Matlab Tool.
F represents the Frequency Domain data and T represents the Time Domain data
Use status to check on the status of these two variables
The following commands are available.
Type 'help command' for help on a particular command
    close  fft    get    help
    ifft   open  plot   quit
    send   status wave
-----
wave> matlab> wave> matlab> wave> matlab>
```



# Additional Notes

- **Pointers are used sometimes in lieu of references because:**
  - Cases when references cannot be used because references are immutable
  - Objects need to be created dynamically
  - To allow classes to be copied via copy constructor
  - E.g. WaveFile: fstream objects inherit from ios\_base and cannot be copied. Thus, a reference or pointer to the object is needed in order to allow for WaveFile to be copied
    - But references are immutable. So the only way is to use a pointer