School of Physics, Engineering, and Technology

# BEng Project Report

# 2022/2023

**Student Name:** Christopher Emmanuel Katzourakis

**Project Title:** Audio On The Blockchain

**Supervisors:** Dr Frank Stevens

Dr Hamed Ahmadi

School of Physics, Engineering, and Technology

University of York

Heslington

York

YO10 5DD

# Abstract

Audio has traditionally been distributed through centralized platforms with control over distribution, pricing and ownership of content. These platforms pay artists very little to stream their material, however due to their audience reach most artists comply with these terms. Blockchain technology offers a means to change this, giving control back to the creators. This project explores the potential of blockchain technology and how it is revolutionising many industries including the audio industry, through the creation of a visual synthesizer which will offer users an easy way of creating NFT (Non Fungible Token) artworks. The audio produced will have a visual element, utilising an audio visualiser to create a visual piece of art that correlates to the music being played. This will then be uploaded to the blockchain and be accessible through an online user interface to mint as an NFT, offering musical and visual artists a means of creation, of both visual and audio artworks, and an instant global marketplace, with sale royalties of their work. The significance of this is research into the ease of use of the blockchain in terms of the regular artist as well as an insight into how audio pieces can work together with an automated visual component to create a unique combination of the two worlds, with a means to generate a revenue source and direct connection between that artist and their audience.

# Contents

# List of Figures

# Acknowledgements

I would like to thank Dr Frank Stevens for his excellent supervision and support throughout the duration of this project. His insight, knowledge and support has been invaluable. I would also like to thank Dr Hamed Ahmadi for his weekly responses to my updates and the feedback he has provided. Finally I would like to thank my mother Pauline Katzourakis for her continuous support in my education, pushing me to pursue my goals as a mature student, ensuring that the report I have produced makes grammatical sense and spending countless hours helping me achieve the best results possible.

# Introduction

This chapter will offer an explanation of the project "Audio on the Blockchain", analysing the use-cases of blockchain technology in relation to audio, and its adoption by industry professionals.

## 1.1 An Overview of Audio on the Blockchain

The emergence of blockchain technology in recent years has been groundbreaking for many industries, offering a decentralized, secure and transparent platform for financial transactions in the form of cryptocurrency and non fungible data storage. One industry where this is particularly prominent is in the field of digital media, where through the use of blockchain technology, the creation distribution and monetization of content can be completely revolutionised.

For years audio creation and distribution has abided by a traditional path. The artists create music, which is recorded through an investing record label. The record label controls distribution of the material through CD, record sales and via centralised streaming platforms such as Spotify and Apple Music, who charge a subscription fee for their users to stream the music.[1] These streaming platforms based on the number of streams the music receives pay a percentage of royalties to the record label, who also take a percentage, and then return the rest to the artist, as shown in figure 1.1. The artists are effectively forced to part with a large portion of the profits from their work in order to reach their audience. Blockchain technology if used correctly has the ability to solve both of these problems, and has currently many large scale musical artists advocating for its use, including popular rappers "Eminem", "Snoop Dogg" and "Pitbull" [2–4]

The project "Audio on the Blockchain" explores how blockchain technology can be used as a means of distribution of recorded audio snippets with visual accompaniment, analysing the positives and negatives of using the technology and the utility that can be assigned to holders of the assets. Through the creation of a decentralised web application and a visual manipulating synthesizer a collection of non-fungible tokens, or as they are commonly referred to, NFTs were minted onto the blockchain as an experiment of how the technology can bridge the gap between artists and their audience offering financial freedom and royalty control over their distributed content.

By exploring these topics the aim of the project is to contribute to a better understanding of the the potential impact that blockchain technology can have on the evolution of the audio industry. Also offering an alternative to traditional means for artists offering better control of their music licensing, consumption and royalty distribution, as well as new forms of fan engagement and participation.

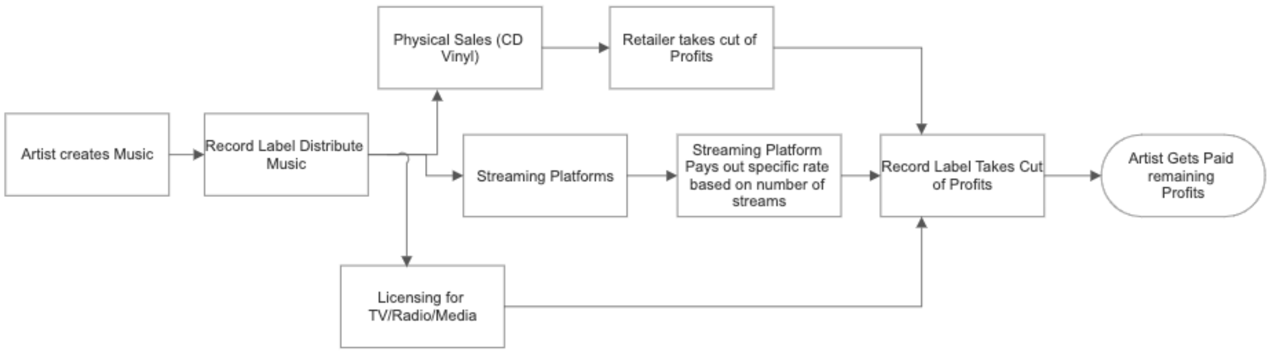


Figure 1.1: Showing a block chart of the traditional methods of music distribution.

# Audio Visual Synthesis

This chapter will explain the choices made in regards to the audio and visual synthesis of the project. It will analyse the process of creating the insturment and implementing the visualiser detailing the objects used and their functionality. An overview of the recording process of the video will also be included breaking down the process from conceptualisation to outputting the final visual artwork.

## 2.1 Conceptualising the Black Hole Sun Synthesiser

The "Black Hole Sun" synthesiser was the main audio component of the project facilitating the audio and visual creative platform.

The creation of a digital synthesizer can be a complex process requiring careful consideration of audio synthesis techniques and a visually appealing and easy to control design. This section will outline the key concepts and challenges involved in designing and building the synthesiser, analysing the code and algorithms used within, including aesthetic considerations and an analysis of the visual component produced to visually depict the audio played.

It was decided that the use of a keyboard styled synthesiser would be the ideal approach to creating this virtual instrument. By offering a familiar and widely used interface for controlling the notes, the synthesiser can reduce the learning curve and increase accessibility of the instrument, making it simpler for the regular user to understand. As the synthesiser was being made within the Max/MSP software this was also easy to implement as the object kslider automatically creates a keyboard with pressable buttons that output the midi numbers of the notes played.

From the initial report justifications had been made to use a combination of FM Synthesis and Additive Synthesis to create the sounds the synthesiser would produce. Consideration had to be given to how these would be implemented within the instrument, together with a single point of user control, or separately allowing the user individual control of the two different synthesis techniques. Combining the two limited the amount of variation that the synthesiser could produce, whereas separately, although linked within one instrument the user would be able to individually configure the two sounds, giving a more unique experience for every individual. Therefore the decision was made to adapt a two-in-one approach, two different synthesisers working harmoniously within one instrument, with two keyboards to control them each separately.

Another key consideration within the conceptualisation of the design was which effects to include within the synthesiser. Each one can dramatically alter the functionality and versatility of the synthesiser so after reviewing a range of different effects and how they can be implemented within a MAX/MSP patch the decision was made to include a delay. This would give the ability to create a sense of space and add depth to the sound as well as offering a way of creating rhythmic patterns and atmospheric textures. The inclusion of a sequencer for the additive synthesiser was also important as it would give the user a way of generating complex and evolving repeating patterns of notes, which they would be able to play over the top of with the more lead oriented FM synthesiser.

The visual element as well as the overall design needed to be eye catching and stand out. For the produced artworks to be appealing and maintain visual attention, it was decided that bright colours were to be used, set on a dark background with a futuristic computer graphic effect, making it look professional and innovative. Every interactive object was to be labelled appropriately for ease of understanding, and the visual product of the audio was to take inspiration from early 2000's Windows Media Player music visualisations, as seen in figure 2.1, making them unique and captivating.

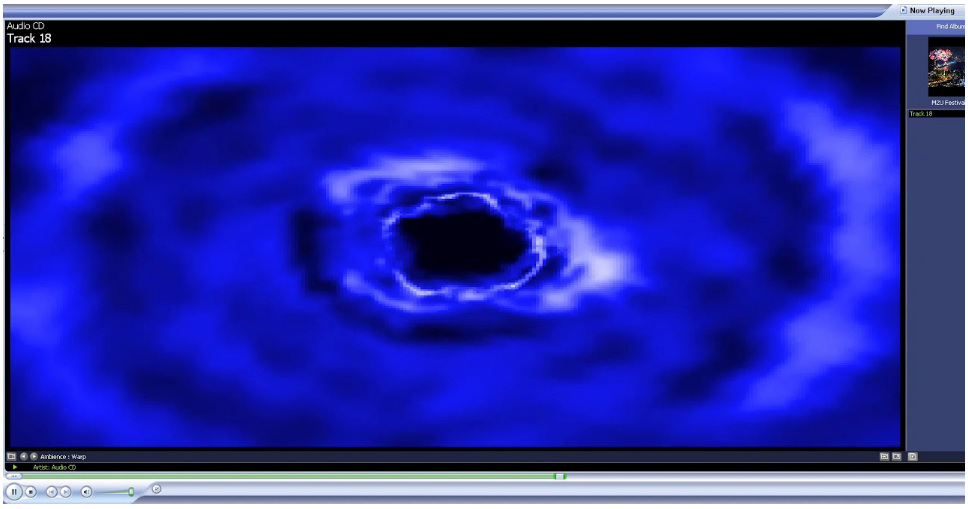


FIGURE 2.1: Showing an example of an early 2000's version of Windows Media Player visualising audio. [5]

## 2.2 The Additive Synthesiser

The process of building the synthesiser was a majority portion of the project. Understanding how to create the different elements of the synthesiser took a lot of consideration and pre-planning. Using the Max/MSP visual programming interface however made the process much easier to understand and with the plethora of information available assisting in its use, it was clearly possible to achieve everything that was decided upon.

Unlike most programming languages Max/MSP is based on the concept of patching together pre-built objects which represent various audio and control functions. These objects are displayed on a graphical interface as a small rectangle and their properties are set by adding a title to them. Depending on the object they typically have inlets and outlets allowing for them to be patched together and send and receive data, using virtual patch cables displayed on the interface.[6] This makes it easy to visualise the flow of data through the patch cables and understand how each object is processing the signals received.

The first step of the creation process was the construction of the additive synthesiser. Additive synthesis works by adding together multiple sine waves of varying frequencies, amplitudes and phases to create a composite waveform that can be shaped to produce a wide range of sounds. This is due to the complexities of the waves produced by the addition of one to another, creating unique characteristics for each new wave produced. Effectively any waveform can be created through the combination of multiple sine waves, as shown in figure 2.2. In the case of a simple harmonic oscillator, the sine waves that make up the sound have frequencies that are whole number multiples of the fundamental frequency. These additional frequencies are commonly referred to as the "harmonics" of the sound. [7]

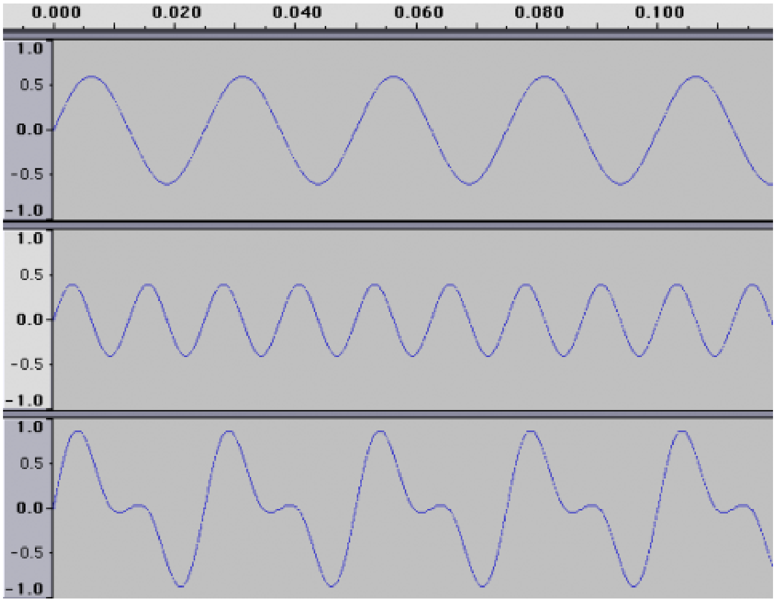


FIGURE 2.2: Showing an example how additive synthesis combines waveforms to create new sounds. [8]

In The Black Hole Sun Synthesiser the harmonic content was very important. Initially a kslider object was used, this object displays as a keyboard on the patch canvas. When keys on the keyboard were pressed a message was sent out of the outlet with the midi number associated with the note of that key. This note was then sent to a mtof (midi to frequency) object, converting the midi number received into a frequency number. This then split off into two paths from the outlet; one leading directly into a multichannel signal object, converting the frequency number into ten channels of signal, and the other into a pak.harmonic object. This object outputs a harmonic series from the fundamental frequency input into the object. It also allows the multiplication of harmonic series, creating a wider range of pitches produced. This multiplication variable was connected to a knob that was adjustable on the user interface, giving the musician an option to change the harmonic sound of the notes that they play. This was also converted into ten channels of signal and both paths were passed through a multichannel cycle object. The cycle object in Max/MSP takes an input and then relays the information to a sinusoidal oscillator. In order to accept the ten channels of signal there needed to be ten channels of oscillators for each of these. They then were passed through a mixdown object effectively adjusting the gain of the oscillators.

An envelope refers to the shape of a sound wave over time. Typically described by four parameters, attack, decay, sustain and release (ADSR). The attack describes the initial transient of the sound, decay the decrease in amplitude after the attack, sustain the steady-state portion of the sound and release the decay once the sound has ended.[9] This is demonstrated in figure 2.3. As the oscillators did not have any envelope information, the audio output was a continuous note. Therefore the synth was configured so that when a key on the kslider was pressed it would also trigger a button or bang. This button would trigger a message relaying ADSR information to a multiplier object. This multiplied the sound by a number between 0 and 1 whilst the information from the envelope was being relayed. Once the information finished it would revert back to multiplying the sound by 0 preventing a continuous note being heard.

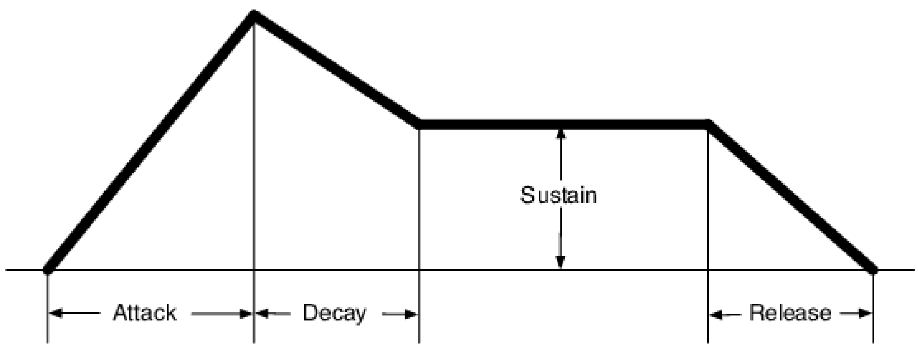


FIGURE 2.3: Showing the elements of an ADSR envelope. [10]

Once passed through the envelope there were two paths that the signal could take depending on the user's selection. If the delay was turned on from the user interface, it would go through a comb filter object with user adjustable values for the delay time and feedback. However if the delay was off, the signal would bypass this. This was achieved through the use of bangs and toggle objects. When the button to turn the delay was pressed this would in turn trigger a toggle. A toggle within Max/MSP is an object that when triggered outputs a constant value of one and when not triggered a constant output of zero. When the delay was turned on the button would trigger a toggle leading to a multiplier object effectively turning on that output. This would also trigger another toggle turning it off and therefore stopping any signal flowing through the other output. When the delay was turned off the opposite would happen, bypassing the comb filter.

In order to add versatility to the synthesiser and allow for more user control a sequencer was added. A sequencer is a tool that allows the user to program and control the sequence and timing of notes being played. This is achieved by sending midi information from the sequencer through the synthesiser at a user adjustable rate.[11] This therefore needed adjustable parameters for the number of notes in the sequence, the speed at which the notes would be played and to turn the sequencer on or off. In order to achieve this the additive synthesiser was copied and similarly to how the delay on/off button functioned through toggles, when the sequencer button was pressed it would not allow the signal to go through the original synth and would effectively turn on the copied synth. This copied version needed to be altered slightly in order to create the sequencer component. The MIDI information that was sent from the kslider object firstly passed through a zl.stream object. This object creates a list from the information input. The length of the list created is defined by the second inlet on the object. By making this user adjustable, through the use of a slider the adjustable parameter for number of notes in the sequence was created. Separately to this a metronome object was created in order to trigger the notes to be played. Based on an input variable speed the metronome object sends ticks through its output. This can trigger other objects at the rate of the ticks, therefore resolving the method of making the speed user-adjustable. The metronome object was connected to a counter object which in turn triggered the list of MIDI notes. The list was cycled through using a zl.mth object which extracted the individual MIDI numbers at the rate set by the metronome. Once the MIDI numbers were extracted the additive synthesiser copy was connected and functioned, cycling through the sequence of notes at the speed set by the user.

## 2.3 The FM Synthesiser

Frequency Modulation synthesis is a audio synthesis technique that was first introduced by John Chowning in the late 1960s and is based on the idea of modulating the frequency of one sound wave by using another sound wave. Two oscillators are used; a modulating oscillator and a carrier oscillator. The modulating oscillator is used to control the frequency of the carrier oscillator, meaning that as its own frequency is changed the frequency of the carrier oscillator is also changed. This creates sidebands, frequencies higher and lower than the carrier frequency. The amplitude of the sidebands depends on the frequency and amplitude of the modulating oscillator. When it is at its highest point, it creates sidebands that are higher in frequency than the carrier frequency and when at its lowest point, sidebands that are lower in frequency than the carrier frequency.[12] This is shown in figure 2.4. In mathematical terms frequency modulation is defined by the following equation:

$$\mathrm{f}(x) = \mathrm{a}(x)\sin(2\pi \mathrm{f}_c x + I(x)\sin(2\pi \mathrm{f}_m x)) \tag{2.1}$$

Where a(x) is amplitude of the waveform, I(x) is the modulation index (or ratio/depth), fc is the carrier frequency and fm is the modulation frequency[13]

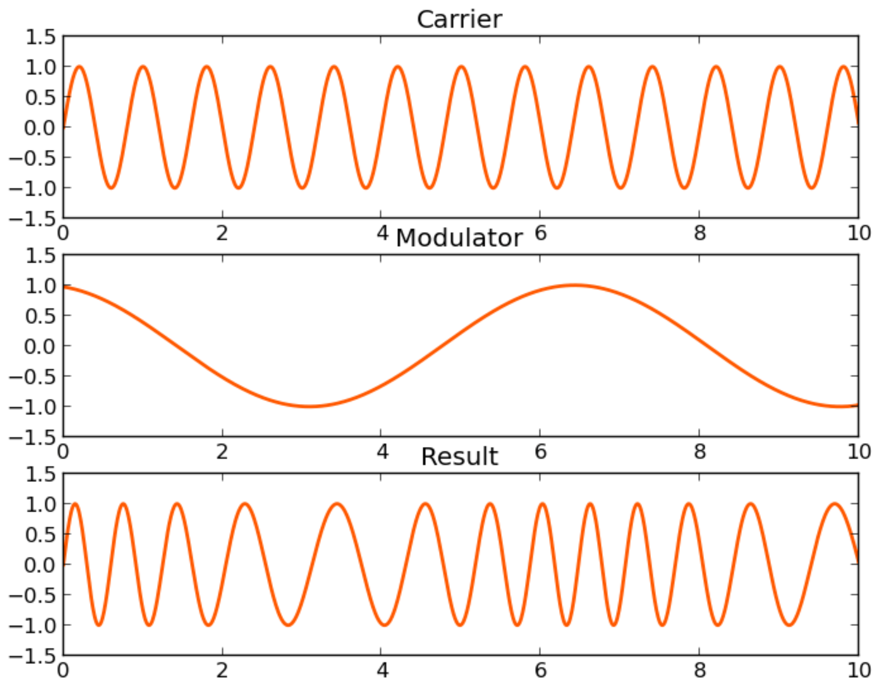


FIGURE 2.4: Showing the carrier frequency, the modulation frequency and the resulting waveform. [14]

In order to create the FM Synthesiser section of the instrument there were again certain details to consider. In order for the synth to be fully versatile and allow the user to control it properly there needed to be adjustable options for the modulation frequency, the frequency depth, and the ADSR envelope. In order to implement this firstly an input was needed. Due to ease of use and familiarity this was chosen to be a kslider object similar to the additive synth. This kslider object sent a MIDI number into the synthesiser which was then converted into a frequency number. Separately to this a live.dial object was created to control the modulation frequency. This dial ranged from 1-1000 and sent the value through to a sinusoidal oscillator. The oscillator would send that frequency into a multiplier object where, again using a live.dial with a range of 1-1000, the signal was multiplied by a user adjustable value. This was then added to the frequency of the initial key played and fed through another sinusoidal oscillator. An ADSR object, Max/MSP's inbuilt object to create an envelope, was then created in order to give the musicians the ability to shape their sound. Dials for attack, decay, sustain and release were connected to this and implemented into the user interface. The resultant information from the outlet of this was fed into a multiplier object, with the output of the second oscillator, applying the envelope to the output of the FM synthesiser. Much like the additive synth this was then split, with one path leading through a comb filter and the other bypassing it. Depending on the user's choice to add delay or not, a toggle system closed one path or the other.

## 2.4 Routing the Audio

Once the additive and the FM synthesiser audio signals had been processed they had to be output in order for the user to hear what they were playing in real time. Firstly however it was decided that a panning section should be added in order to give the user control over how they would like their mix of the two synthesisers to sound. Panning is the distribution of a sound signal across the stereo field. It is a technique used to create the illusion of space and dimension in a mix, and it enables the placement of individual sounds in specific locations within the stereo image. The pan control adjusts the balance of the audio signal between the left and right channels, allowing the sound to be placed anywhere between the two channels. In order to recreate this in Max/MSP a combination of sine and cosine objects were used. The signal was fed through a left and right inlet which led to a multiplier object. A live.dial ranging from 0-1 was added to the user interface and the value of that was multiplied by 1.5707. This was then split and connected to a cos and sin object. When the value of the dial was at 1 the cosine of 1.5707 was 0 and the sine was 1. These connected to each of the multiplier objects and based on the value of the dial would lower the volume of one side or the other. Once through the panner the signal was sent to a live.gain object. which effectively works as a volume fader, giving the user control of the individual gain of each synthesiser. From their own individual live.gain objects the audio signals were then routed to a combined live.gain object, gathering all the audio and giving an extra point at which the user could adjust the gain exiting the synthesiser and from there the output was sent to an ezdac object. The ezdac object sends whatever audio it receives to the audio output that has been selected of the computer. However as well as being routed out of the audio output this combined signal was also routed to the visualiser, giving the audio information needed to create the visual artwork of the instrument.

## 2.5 The Visualiser

The visual element of the project was by far the most complex. Understanding how to create visualisations of audio and display them as a piece of art required a lot of research as well as a high level of understanding of the jitter elements of Max/MSP. In the initial project plan it was decided that a software called GEM would be used to develop the visualisation of the audio, however it was quickly discovered that there were several compatibility issues between GEM and Max/MSP. Initially being developed for Pure Date, a similar software to Max/MSP but effectively a light version, without as many capabilities, it was very outdated, having not been updated since 2019. This and the fact that the only option to use GEM with Max was by using a third party port, meant the decision to use GEM was rejected.

Jitter is a toolkit of objects for working with video, matrix data and other time-based media within Max/MSP. It is an extension of the Max/MSP programming environment and offers objects that can manipulate and process video and image data in real time. The matrix processing capabilities made it the obvious alternative to GEM, as well as its seamless integration with Max/MSP and its ability to create 3D visual processes in real time. [15]

In order to create visuals from audio data an FFT (fast fourier transform) must be completed on the incoming signal. This transform converts the signal into the time and space domain, a representation of which is included in figure 2.5, allowing it to calculate the DFT (discrete fourier transform) of the signal as Cartesian coordinates (Otherwise known as Rectangular coordinates. These relay the position of each wave in a three dimensional space, giving a positional value to visualise the signal being played through the synthesiser.[16] The Discrete Fourier transform is defined as follows: [17]

$$\mathrm{X}_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n * e^{-i2\pi kn/N} \tag{2.2}$$

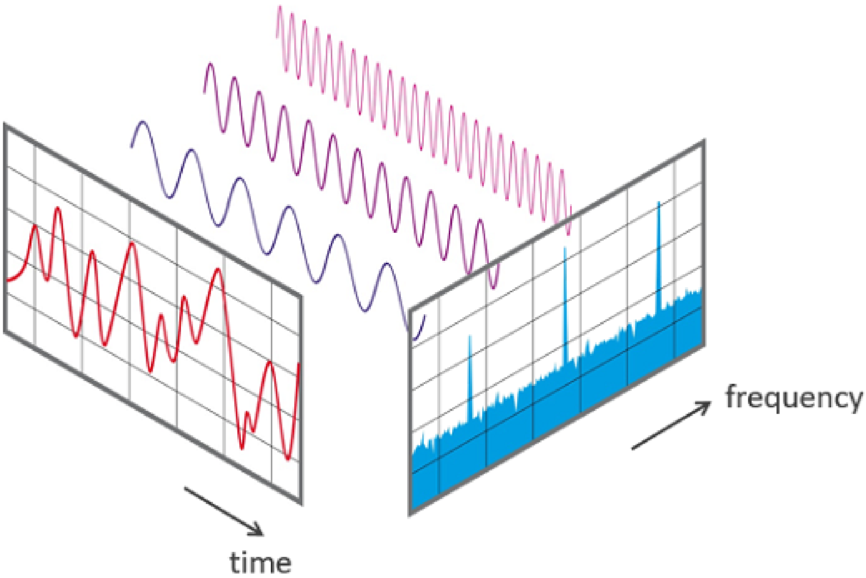


FIGURE 2.5: Showing a visualisation of a waveform in the time and frequency domain. [18]

Once the FFT has been completed and the Cartesian coordinates received, using a cartopol object in Max/MSP and feeding the real and imaginary values to it these can be converted into polar coordinates. Polar coordinates represent a point in a two-dimensional space using a distance and an angle. The distance (r) represents the radial distance from the origin to the point or in this case the amplitude of the signal, while the angle ($\theta$) represents the counterclockwise angle measured from the positive x-axis to the line connecting the origin and the point. Together, r and $\theta$ uniquely identify the location of the point in the polar coordinate system. The difference between rectangular and polar frequency domains are shown in figure 2.6

A matrix was then created using an fftinfo object. This object analyses the FFT and outputs data such as the Frame size of the FFT and the spectral frame size. The spectral frame size

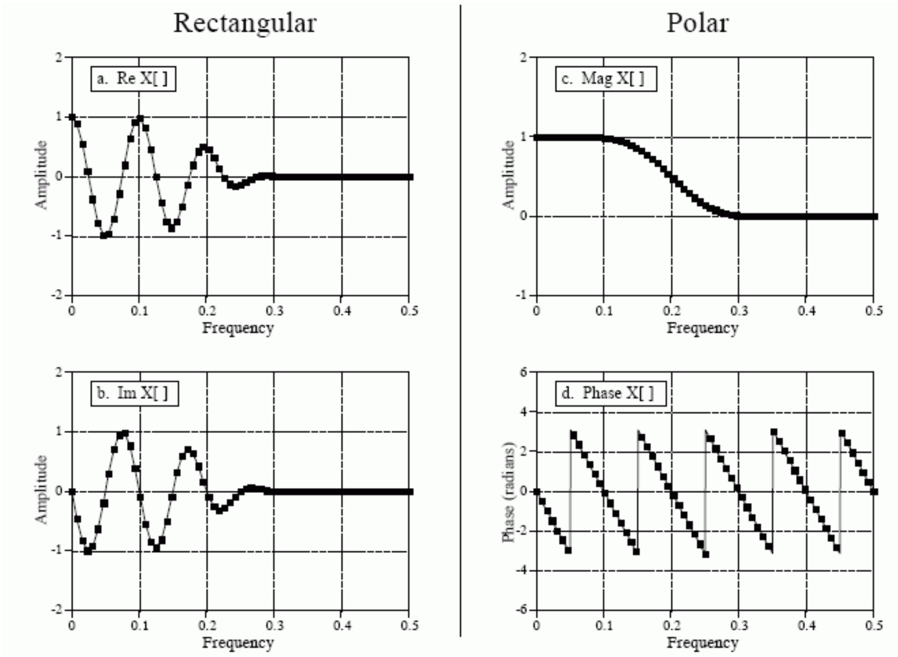


FIGURE 2.6: Showing the differences between a signal in the Cartesian (Rectangular) and Polar frequency domains. [19]

refers to the number of samples of an audio signal that are processed at once to compute the frequency spectrum of the signal. Using this in collaboration with a dimension object the dimension of the matrix was changed to the size of the spectral frame size. The incoming amplitude data from the polar coordinates was then sent to a jit.poke object which wrote the amplitude, a constant 0 signal and the FFT index to the matrix. An inlet toggle controlled how regularly the FFT would be calculated. Having the toggle on would make the FFT function continuously whereas if turned off, it would not do the calculations and therefore effectively turn off the visual component of the synth. This option was added to the user interface giving control to the musician as to whether they wanted to use the visual component or not. The matrix with the frequency data was then inverted on its y axis in order for the visualiser to show the highest frequencies at the top of the window and the lowest at the bottom. Connecting a visual window object to this displayed a snapshot of the frequency incoming into the visualiser. However in order to expand the artwork beyond single snapshots a scrolling view of the frequency data as it was incoming was created. In order to do this a new jitter matrix was created. Another instance of this matrix was created and connected to a jit.convolve object. This object computes the correlation between two matrices; the initial matrix and a kernel. The kernel was created as a 3x3 matrix and a value of 1 was sent to the left middle cell of the matrix. When jit.convolve combined the kernel and the initial matrix this caused each cell to copy the cell to the left of it. In order to make this scroll, a feedback loop needed to be created by connecting the output of the jit.convolve object back into the initial matrix. When new values enter the matrix they are copied to the left side, creating a real time scrolling data effect. By connecting the inverted FFT to the matrix and specifying, using attributes of the matrix, that the information should be sent to the leftmost cells, connecting a jit.world object, which compiles and outputs all of the jitter objects data, to trigger the matrix to update at the same rate as the frame rate of the visuals resulted in a smooth flowing scrolling view of the frequency domain.

Since the scrolling view was completed this then had to be projected onto a mesh. The jit.gl.mesh object creates a geometric surface from the information provided from the jit.matrix that is connected to it, displaying the spacial coordinates into a three dimensional setting. However in this state the matrix would only display as a singular diagonal line as it was two dimensional. Using a jit gen object this issue was resolved. The jit.gen object allows the user to create custom image processing effects using GLSL code (OpenGL Shading Language). This gives the ability to manipulate matrix data in real time, adding colour manipulation, filtering and creating complex visual patterns based on the data input. The operation of jit.gen would process every cell of the matrix simultaneously. Inside the jit.gen object a torus operator was created connected to swiz.x, swiz.y and swiz.z objects. The torus operator sends information to the swiz objects to create a torus shape in the 3 dimensions. The swiz objects extract the x, y and z components being sent to them. This was then sent to a vector object displaying the torus shape in a 3 dimensional mesh. In order to add the frequency data from the matrix to this three dimensional object, it was passed through an inlet into the jit.gen object. In order for the data not to disfigure the torus completely it was multiplied by 0.25 and subtracted from the z axis leading into the vector. This was then relayed to the jit.gl.mesh object and projected the frequencies onto the 3 dimensional object creating the patterns displayed in the visualiser. For the colour selection the frequency data from the matrix was amplified by a significant factor and connected to the colour array inlet of the jit.gl.mesh object. This displayed the data on the sphere in a white colour.

Through trial and error it was discovered that if passed through different vectors and connected to the colour array the torus object and the audio data would be displayed with different colours.

After copying the jit.gen object multiple times using different vector combinations a total of 16 different colour schemes were created. Creating 16 different buttons for the different colour schemes available to be selected on the user interface, these were each individually connected to their own bang objects. When selected the bang would activate the jit.gen object of the colour scheme combination required and in turn trigger the correct path for the switch objects to select. Switch objects accept two inputs; based on a message of either 1 or 2 they output the respective input. Through this elaborate switch maze created, when selecting the button the switches would all be correctly aligned to allow only that jit.gen object to reach the colour array input of the mesh.

In order to control the angle at which the visualiser displayed the torus object and frequency data a jit.gl.handle object was implemented. This object allows the user to rotate the mesh and view it from different angles, as displayed in figure 2.7. In order to create the best visual result an angle was selected that visualised both the object and the frequency data clearly. Through the use of attributes of the jit.gl.handle object, the viewpoints position, angle, zoom and rotation were all locked into place. This would avoid any confusion with displacing the mesh and effectively breaking the visualiser, by removing the capability of the user to alter these parameters. Everything was compiled through the jit.world object created previously and displayed on a jit.pwindow object; an object that visualises the information received as pixel data and displays it visually, creating the artwork based on the audio passing through.

## 2.6 The User Interface

In order to fill the criteria of the project the synthesiser would need to have a well designed user interface, giving the user a significant amount of control over the sound they produce. The visualiser and audio output also needed to be recorded and saved to the computer as a consolidated mp4 file.

Having already established the components that would be available through the user interface the task was to assemble them in a way that would be appealing to the eye and also easy to use. Max/MSP has two ways of displaying its code. In patching mode the user is able to

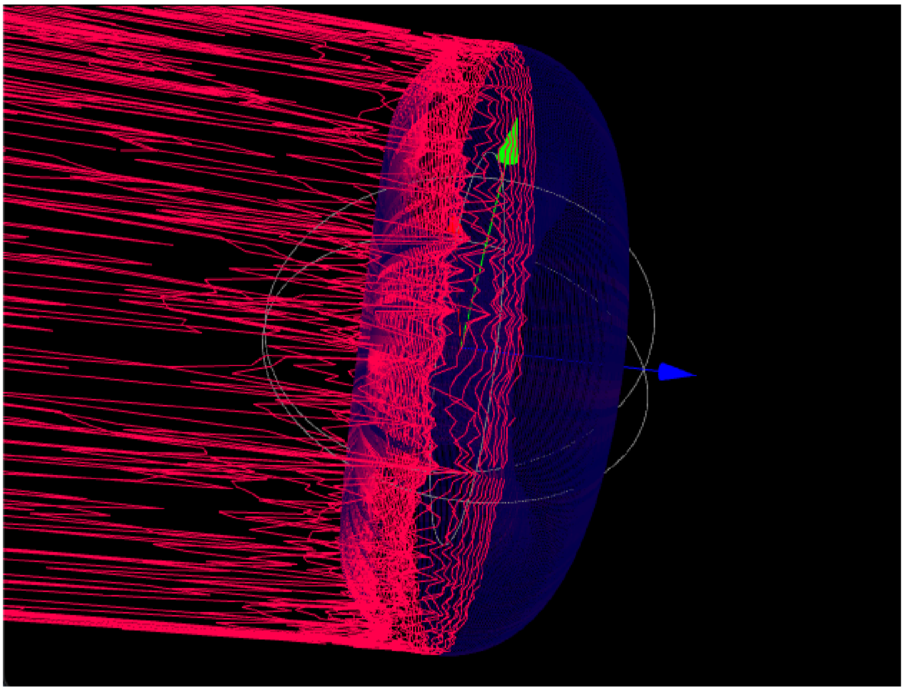

FIGURE 2.7: Showing a sideways angle view of the 3 dimensional torus object with audio being relayed through it.

see all the connections between objects and view items not intended to be seen by the final user, presentation mode however allows the display of specific objects that the programmer would like the user to see and interact with. In order for the synth to launch automatically into presentation mode a loadmess object was used with the prompt to display the patch in presentation mode. This forced the instrument, upon loading to automatically display in presentation mode.

In order to consolidate the files together several approaches were tested in order to establish the most efficient means of recording. The initial intention was that all processes would be handled within Max/MSP however after many attempts this proved impossible. Using a package called "Syphon" the video could be sent to a server and recorded within a companion program "Syphon Recorder". In order to record the audio a virtual sound card needed to be installed on the system. This allowed the audio to be routed directly into the "Syphon Recorder" and recorded with the video. As the output of Max would be set to the virtual soundcard however Audacity, a free DAW software, was used for monitoring purposes, routing the virtual soundcard output to the inbuilt audio device. The issue with this approach was that "Syphon Recorder" did not have the option to change the format of output and would only output .mov video files. As the NFT section of the project required specifically mp4 files, the mov files would need to be converted manually outside of the project. In order to bypass this process it was decided that OBS, another video editing software with the capability of outputting mp4 files, would be used to record the video. Fortunately OBS also recognised the "Syphon Server" as a video input and was therefore easy to integrate. The user would simply need to press record on OBS, play music on the instrument which would output an image and then press stop recording on OBS to output the video file. A visualisation of this is shown in figure 2.8.

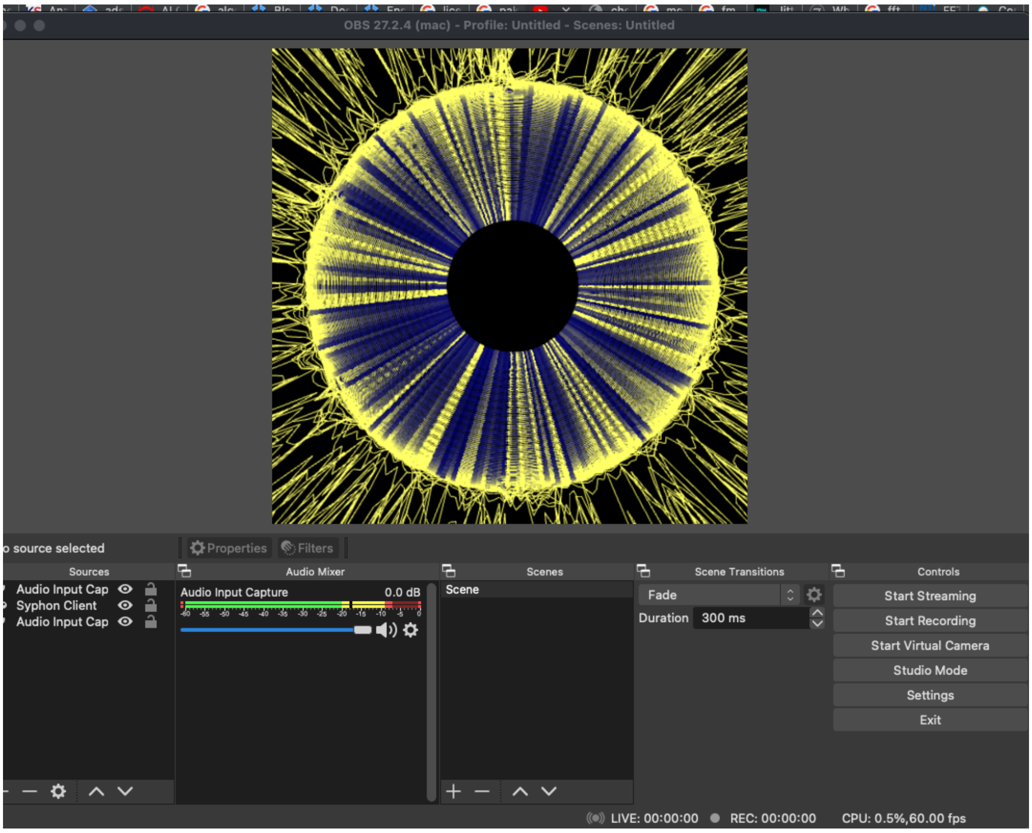


FIGURE 2.8: Showing an example of a piece of artwork being recorded with the audio in the OBS software.

Taking inspiration from old school Windows Media Player the theme was a contrast of bright colours on a dark background, reminiscent of vintage science fiction films such as "Tron". The centrepieces of the interface needed to be the keyboards and the visualiser, as those were the key elements of the synthesiser. They were therefore placed centrally and expanded in size. To differentiate the elements of each synth, a colour coordination was chosen, with a "Star Wars" inspired colour scheme. The FM synthesiser keys and dials were displayed in red and those of the additive synthesiser in blue. The visualiser was placed above the keys as a centre console for the layout and the delay buttons and controls were displayed to the left and the colour changing vector buttons to the right. These matched the colour of the combination of colours they would change the visualiser to. To avoid confusion this layout also separated them from the audio elements of the user interface. The background image and labels for each button were created in Adobe Photoshop and imported to the patch. By taking a screenshot of the layout of the elements aligning the buttons was simplified, creating darker sections on the image to emphasise them. The title "Black Hole Sun", (inspired by the famous Soundgarden song as well as the visuals resembling a black hole), was placed at the very top of the frame completing the instrument. The completed user interface is shown in figure 2.9.

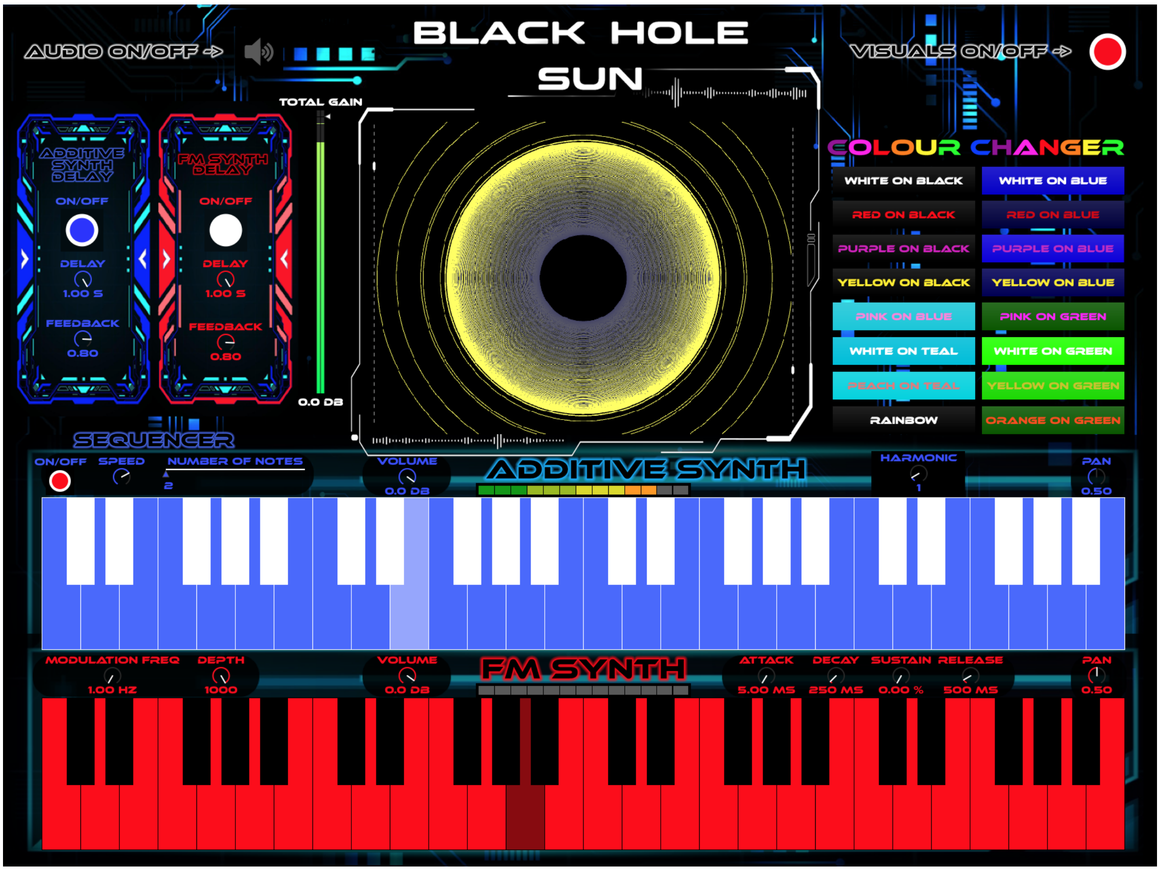


FIGURE 2.9: Showing the final layout and colour scheme of the instruments user interface.

# ...On The Blockchain

This chapter will analyse the blockchain component of the project, explaining the differences between several blockchains and how they function. It will also detail the process followed in order to deploy the smart contract onto the Mumbai chain and create the decentralised application, used to interact with this.

## 3.1 What is a Blockchain?

A blockchain is a decentralized, digital ledger that records transactions in a way that is secure, transparent, and tamper-proof. It is essentially a database that is managed by a network of computers, rather than a central authority. The network of computers work as validators for the transactions on the blockchain ensuring that each transaction is legitimate. In the case that one validator nefariously confirms a different set of transactions this result will be invalidated due to the rest of the network confirming the correct ones. Each block in the chain contains a number of transactions, and once full, a block is added to the chain. From this point on it cannot be altered or deleted and remains as a permanent record on the blockchain, visible by all through a blockchain explorer.[20] The reward given to validators for confirming these transactions is in the form of a token. These tokens offer a financial incentive for the validation as they work as a currency by users of the blockchain to submit their transactions. Much like assets on the stock market these tokens fluctuate widely in price and have varying values depending on the supply of tokens, the integrity and the popularity of the chain. These tokens are commonly referred to as cryptocurrencies.[21] Different chains of blocks have different characteristics, some of the most prominent of which are Bitcoin, Ethereum and Polygon.

Bitcoin was the first blockchain to be created establishing the technology and leading to the creation of many more chains. However the technology is now proving slow and somewhat outdated. Bitcoin mainly functions as a transfer of financial transactions limited by its network only being able handle a total of 7 transactions per second. The validation process used functions on a proof-of-work algorithm. This algorithm allows for the currency Bitcoin to be "mined" by means of creating a virtual competition. Computers on the network compete against each other, solving many highly computationally-intensive mathematical problems. The first computer on the network to solve all of these problems validates the transaction and is awarded with a portion of the currency. This approach however results in many computers failing to validate but expending a lot of energy in the process. This has costs both in electricity used and in wasted energy, contributing to global warming[22]. The differences between how bitcoin functions compared to regular financial institutions is demonstrated in figure 3.1.

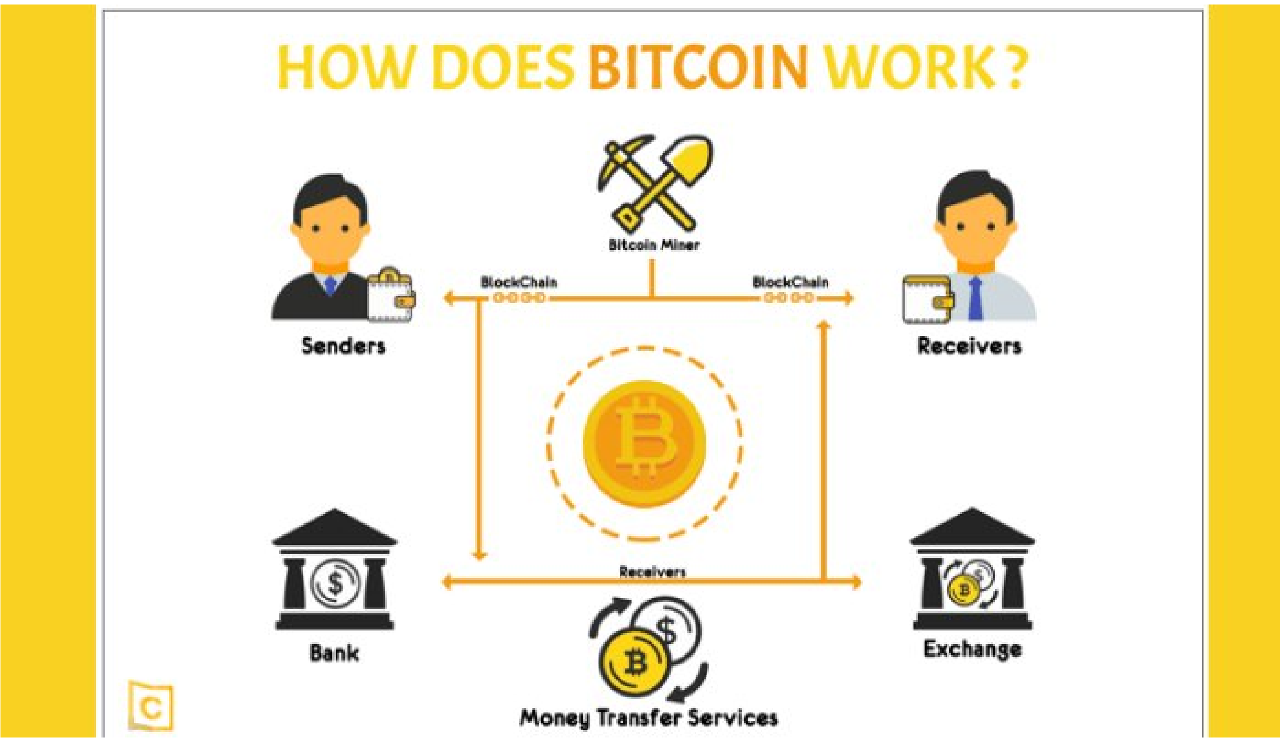


FIGURE 3.1: Showing how bitcoin functions in relation to centralised banks.[23]

Ethereum being a newer updated blockchain, functions differently to bitcoin in the sense that it provides the ability for anyone to create rules for ownership and formats of transactions. This is achieved through the use of smart contracts; a set of rules that are implemented if certain conditions are met and verified. These allow for immediate execution when the terms are met and are completely automated, meaning that no physical records of these transactions exist, other than the information stored on the blockchain. They are secure and allow for the transfer of assets from wallet to wallet without the need of a third party.[24] Ethereum works on a proof-of-stake process. This chooses validators for its transactions based on a stake of their token Ether. Users can opt in by locking up an amount of tokens, giving them the right to validate transactions without competition and removing the negative effects on the environment. However the issues with ethereum are the cost of transactions and the speed. Ethereum can only process 20 transactions per second and the cost of 1 Ether currently stands at approximately £1500. When the network is busy the transaction cost increases in order to give priority to the highest spending users. This can result in the networks use being very expensive.[25]

Polygon is a Layer 2 scaling solution for Ethereum that aims to solve the problems of high transaction fees and slow speeds. It achieves this by providing a parallel network to the Ethereum mainnet, which allows for faster and cheaper transactions. This parallel network is effectively built on top of the Ethereum network by creating multiple sidechains that handle a large number of transactions independently. They are connected to Ethereum and therefore all the transactions are secure and valid. Each sidechain is responsible for a certain set of transactions which increases the overall throughput of the network allowing for speeds of up to 7000 transactions per second. This results in very minimal fees to utilise the network and was therefore chosen as the chain to use for the project.

Most blockchains have a mainnet and a testnet. The mainnet is the real network validating transactions with the primary cryptocurrency, whereas the testnet is a test space, using test tokens to validate transactions and build smart contracts in order to test their functionality. For this project the Polygon testnet, also known as Mumbai, was used in order to avoid the use of real money and operate freely to ensure the system functioned correctly. This could be launched to the mainnet with very little editing to the code.

## 3.2 Blockchain Technology in Relation to Audio

Blockchain technology and audio have developed in combination significantly in recent years. With the development of NFT (Non Fungible Token) technology becoming so prevalent many people underestimate its functionality and how it can be implemented into the distribution of audio. Currently there are many ways to upload audio to the blockchain. Opensea, the largest marketplace for NFT transactions, allows users to upload a variety of different formats including wav, mp3 and ogg files, offering the ability to mint audio snippets as NFTs. This approach has been adopted by many mainstream artists including Grimes, an electro-pop artist, who recently sold over 6 million dollars worth of NFTs. These included a one of a kind video, set to an original song by the musician. This was the only original copy of the music available and sold for almost 400,000 dollars[26]. One of these NFTs is shown in figure 3.2.

FIGURE 3.2: Showing an example of an NFT created and sold by the artist Grimes.

Another popular band Kings of Leon, decided to release their latest album in the form of an NFT, with three different token types. One included a special album package, the second, free concert tickets and the third, a piece of audiovisual art. As early adopters of the technology and with bad publicity circulating about NFT technology in regards to scamming, they faced criticism over this decision, with many users opting to boycott the album due to the innovative medium. [27] However due to the adoption of NFT technology by so many prominent artists in the musical world, the technology and consumer base for audio in this medium is only expanding. Therefore it was decided to incorporate and develop the technology within this project.

## 3.3 Creating the Smart Contract and Deploying the dApp

The specification for the project was to deploy a smart contract which allowed the user to upload their audio visualisation created by the "Black Hole Sun Synthesiser". The user would then be able to mint the result as an NFT using a user interface that interacted with the blockchain, called a dApp or decentralised application. In order to complete these objectives it was decided that thirdweb would be used. Thirdweb is an all-in-one toolkit that allows users to build and mint NFTs as well as building and deploying smart contracts. [28]

Using Visual Studio Code it was very simple to deploy a smart contract. Firstly it was necessary to install thirdweb on the system. This was done by running the command "npx thirdweb@latest". Once installed the option was given to use Hardhat; an open source development environment for building and testing smart contracts. Then an option for different types of contracts was displayed. For NFT contracts on the ethereum blockchain an ERC721 contract required would be. This specifies the rules that the smart contract must follow in order to allow the creation and trading of non-fungible tokens. Subsequently a list of extensions was offered. Selecting "drop" here allowed the contract to utilise a mint and claim procedure which allowed any user to connect their wallet and mint and claim the token available. This then created a template project necessary to deploy the smart contract, initialising the software development kit or SDK and installing all dependencies needed for the project. A solidity file of the contract was created which could then be configured to a specific name which would relate to the contract deployed on the blockchain. Then by simply typing "yarn deploy" the contract was compiled and uploaded to the thirdweb servers and a url was given to access the deploy function in thirdweb.

A crypto wallet then needed to be created. This is a custom wallet with an address where cryptocurrency can be stored and access is given only to the user who created it through a secret passphrase. The chosen wallet was a metamask wallet due to its popularity and compatibility with many platforms. Once created the testnet token for Mumbai; polygon needed to be claimed. This was done by searching for a Mumbai faucet; a website that transfers free tokens to the user, and entering the wallet address. The token would then be automatically sent to the wallet.

When selected, the url that was produced to deploy the contract offered certain contract parameters, as shown in figure 3.3. Firstly the name of the NFT collection needed to be input, in this case “BlackHoleSun”. A symbol for the collection or acronym then needed to be chosen which was selected as “BHS”. Then the percentage of royalties needed to be selected. This would define the percentage that would be deducted from future NFT sales within the collection and given to the artist. In this case 10 percent was selected, adding the wallet address of the newly created wallet to the parameter defining the recipient of royalty payments and primary sale payments. Below this a drop down list for the network selection was available in which Mumbai was selected. The wallet then needed to be connected by pressing the “Connect wallet” button which once connect changed to “deploy now”. This triggered two transactions within the wallet. One deployed the contract and the other connected it to the thirdweb registry allowing the thirdweb account to view all transactions being made on the smart contract and compiling all the information in an easy to use website. Once these transactions were processed the smart contract was deployed and verified on the Mumbai testnet.

Once the contract was deployed a window appeared with the thirdweb account connected to the wallet. Here there was an NFT tab which offered a way to upload the video produced by the “Black Hole Sun Synthesiser”, as shown in figure 3.4. By selecting the “single upload” button a window appeared prompting the upload of a file from the computer. Once the video was selected a name needed to be added, as well as the option of a description of the NFT and properties that were included. This allowed for specific traits from the video to be defined. Clicking the “Lazy mint” button below triggered another wallet transaction to upload the video to be used as metadata for an NFT to be minted. Then claim conditions needed to be set. This would specify the conditions through which the NFTs could be minted. A start time for the sale of the NFTs was available as well as an option to select the price charged in cryptocurrency to claim an NFT.

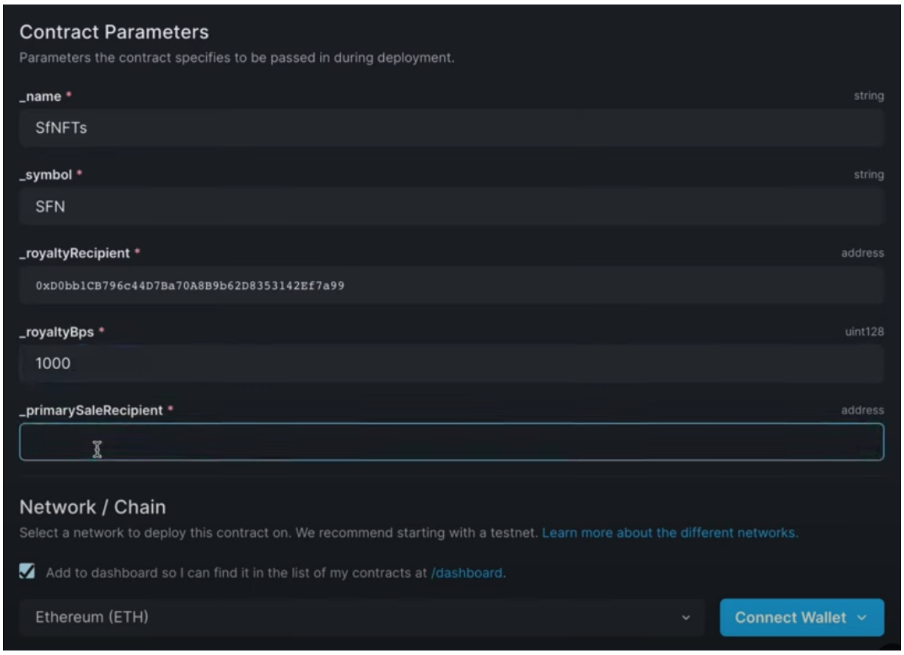


FIGURE 3.3: Showing the smart contract parameters window that allows the user to specify the recipient of funds generated from NFT sales and roylties as well as giving the choice of network to be deployed on.

There was also an option to specify how many can be claimed per wallet and, if necessary, limit the sale to a specific list of wallets. there was also an option to change the currency used to mint the NFTs, but in this case MATIC, the token of polygon was chosen. Once these parameters were set the "save claim phases" option was selected and another transaction was triggered in order to implement these rules into the contract.

Finally the dApp needed to be created and deployed in order for users to be able to interact with the smart contract and mint the NFTs available. This was completed in Visual Studio Code. Selecting a new destination folder the command "npx thirdweb create" was written. This offered the choice to either create a new contract or an app. On selecting "app" it prompted the user to input the project name. Once the name was input a prompt appeared for the framework that would be used. A framework is a prebuilt set of tools libraries and components that can be used to build the application easily. React App was selected and then Javascript was chosen as the language to be used. A new project template was produced based on these selections, installing all dependencies necessary. The App.js file of the template was edited replacing the ConnectWallet button with a Web3Button, giving it the ability to trigger the wallet network to be changed, avoiding crashes. The contract address was attached and the specification of

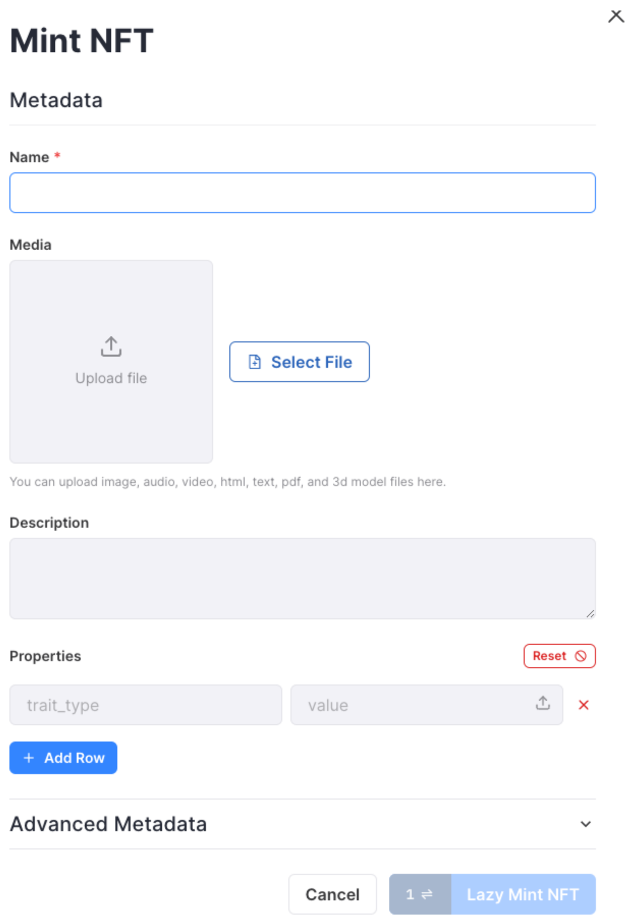


FIGURE 3.4: Showing the interface within thirdapp used to upload NFTs to the smart contract and specify the name, description and traits of the NFT.

the amount of NFTs that could be minted per transaction was set to 1. Once saved, using the "npm build" command a template website was created with the Web3Button imported. When pressed this triggered the metamask wallet to sign a transaction allowing for its interaction with the smart contract. In the templates index.js file the code specified that the wallet would connect to a mainnet network. This was changed to the Mumbai network, so when pressed, if the metamask wallet was not set to Mumbai, the Web3Button button would trigger it to change network. The background was then set within the templates Home.css file to an array of still visualisations produced by the synthesiser and the header text was changed to a hyperlink connecting to the NFT collection on Opensea. The text below this was altered to explain how to mint an NFT and once the wallet was connected pressing the Web3Button would trigger the wallet to confirm a transaction. This transaction would mint an NFT of the Black Hole Sun Synthesiser into the users wallet. An image of the final user interface is shown in figure 3.5.


FIGURE 3.5: Showing the interface used to mint NFTs from the Black Hole Sun NFT collection.

# Evaluation

This chapter will evaluate the effectiveness of the project, assessing its ability to meet the objectives set out to achieve. It will examine the results of the creation of the synthesiser instrument and an analysis on how well the blockchain component was implemented. It will also overview feedback received from testers and future development of the project.

## 4.1 The Synthesiser

The specification of this project was to create an instrument incorporating FM and Additive Synthesis that would visualise the audio as a piece of artwork that could be recorded onto the computer. With the creation of the "Black Hole Sun Synthesiser" all of these points were achieved successfully. FM and Additive synthesis were utilised in the two separate synthesisers that function in collaboration with each other, creating interesting complex sounds that could be played to create unique pieces of music. They both had many adjustable parameters available to the user, clearly labelled as to their functionality, that offered the ability to construct and morph the sounds of each synthesiser. The addition of a sequencer allowed for the additive synth to play a sequence that could be added to via the FM synthesiser and the addition of the Delay modules produced a spacial element to the sounds created. The visualiser functioned better than anticipated, producing interesting patterns and a complex piece of art which could be adjusted in colour, based on the user's preference. Overall the synthesiser was successful in achieving all the targets outlined before its creation.

## 4.2 The Blockchain Component

The specification for the blockchain component of this project was to create and deploy a smart contract and a means to uploading the artwork to the blockchain. Also it was specified that a user interface was to be created for users to be able to interact with their wallets and mint the work as NFTs. This was all achieved successfully with the help of the thirdweb toolbox. The smart contract was deployed successfully onto the Polygon testnet, in order to avoid the spending of actual cryptocurrency and test the process. In future through several small edits the contract could be deployed onto the mainnet producing a true revenue stream for the artist and an easy way of controlling the distribution of their creative output, whilst also benefiting from future transactions through royalties of future sales of their work as NFTs. The utility of holding the NFTs would be completely up to the individual artist, offering IP rights to holders or exclusive content. The choice of the Polygon blockchain worked well in order to reduce costs of smart contract interactions, deeming them almost redundant as the cost for each interaction was fractions of a penny. In terms of user-friendliness the user interface was incredibly simple and easy to use and worked efficiently to interact with the contract and mint the artworks as NFTs into the users wallets.

## 4.3 Future Development

Although this project has now come to its conclusion there are several elements that could be developed further, given additional time and resources.

Feedback of the effectiveness of the synthesiser, that was received based on testing by five students, was to include an option to either save the adjustments made to the sounds as a preset or include pre-built preset sounds that would make the production of quality sounds simpler. In addition the recording process was found to be complex due to the incorporation of external software. If developed further a button could be integrated within the Max/MSP patch triggering the OBS record and stop record function. This would allow the user control from within Max/MSP and avoid interaction with the software running alongside it. Allowing for the change of the 3D shape used within the visualiser could also have been an added function. This would offer a wider variety and more user control in terms of the artwork produced.

In regards to the blockchain component, with some Javascript editing the dApp could also include links to the social media content of the artist, further promoting engagement and popularising them. In terms of interaction for the artist a break down of instructions to upload and mint the NFTs could be provided in addition to preset Javascript packages. These would allow the user to change the theme and background of the user interface without the necessity of coding skills. A custom URL could also be purchased to make finding the mint link more accessible.

## 4.4 Time Management of the Project

In the initial proposal of this project a specific time schedule in the form of a Gannt Chart was provided as displayed in figure 4.1. This was altered slightly due to unforeseen difficulties with certain components of the project overrunning. Certain aspects were also found to be unachievable using the proposed software and therefore changes needed to be made. However thanks to the use of online resources and a great deal of trial and error, the project was completed in good time. In future projects this could be avoided through further research being completed before the project start and distributing the time allocated to the complex and straight forward elements of the project more appropriately. However the final project was completed and functioned as intended within the time constraints set.

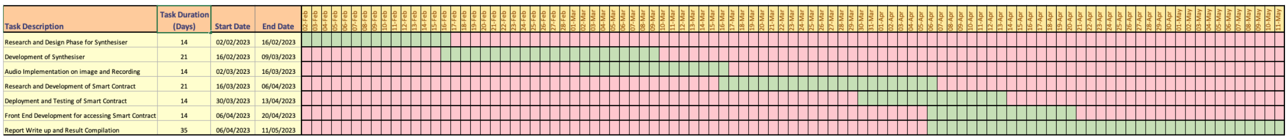


FIGURE 4.1: Showing the Gannt chart produced to time manage the project.

## 4.5 Conclusion

Audio on the blockchain solves a variety of issues within modern day audio distribution. Placing control of distribution and content accessibility back into the hands of the artist rather than large multimedia distribution companies. The visualisation of audio offers a means to artists of visual and audio production of content within each others realms. This would extend the opportunity to artists with physical disabilities within the audio and visual range to produce content based on either the visualisation of the audio frequencies or sounds produced. This synthesiser offers a means of bridging the gap between the different mediums of art. Each element can be used separately, using the synthesiser as an instrument alone, or by disconnecting the visualiser from the synthesiser and feeding in directly pre-produced music in order to offer a visual element to it. The importance of this project lies within this automation of artwork as well as expanding the understanding of how blockchain technology can be utilised within different mediums. Many think of blockchain as purely a transfer of wealth however this project aims to prove that there is much more scope for the uses of this technology. With further advancements blockchain could truly become part of the mainstream within audio distribution.

# The Black Hole Sun Synthesiser

Within the files handed in with this report are several folders. The folder "Black Hole Sun Synthesiser" includes all the files associated with the creation of the "Black Hole Sun Synthesiser". This includes the Max/MSP patches which contain all the code written and structure of the instrument.

# The Blockchain Component

Within the files handed in with this report are several folders. The folder “dApp” includes all the files and code written in order to deploy the decentralised application which interacts with the smart contract. The folder “black-hole-sun-contract” contains all the files including the code written and created in order to deploy the smart contract onto the blockchain.

# Bibliography


[1] H. McDonald, "What is music distribution?" Aug 2020. [Online]. Available: https://www.liveabout.com/music-distribution-defined-2460499.

[2] T. Luno, "How crypto technology is changing the music industry," Jan 2023. [Online]. Available: https://discover.luno.com/how-crypto-technology-is-changing-the-music-industry/

[3] C. Guaramato, "Eminem promotes a bayc video alongside snoop dogg," Jun 2022. [Online]. Available: https://www.cryptopolitan.com/eminem-promotes-a-bayc-video/

[4] J. Santo, "Pitbull is turning to blockchain in hopes of saving the music industry," Apr 2018. [Online]. Available: https://www.cnbc.com/2018/04/24/pitbull-is-turning-to-blockchain-in-hopes-of-saving-the-music-industry.html

[5] *Windows Media Player 9 Series Visualization (Ambience Warp).* Note10 AuraRed Galaxy, Dec 2019. [Online]. Available: https://www.youtube.com/watch?v=awglgBJohtE

[6] V. J. Manzo, *Max/MSP/Jitter For Music: A practical guide to developing interactive music systems for education and more*, 2nd ed. Oxford University Press, 2016.

[7] G. Reid, "An introduction to additive synthesis," Jul 2004. [Online]. Available: https://www.soundonsound.com/techniques/introduction-additive-synthesis

[8] "Additive synthesis." [Online]. Available: http://musictechmusician.weebly.com/additive-synthesis.html

[9] A. V. Buuren, "Adsr envelopes explained: 4 stages of an adsr envelope - 2023," Jun 2021. [Online]. Available: https://www.masterclass.com/articles/adsr-envelope-explained

[10] T. Kvifte, *Instruments and the Electronic Age: Toward a Terminology for a Unified Description of Playing Technique*, 01 1989, vol. 35.

[11] L. Mitchell, "What is a sequencer in music? how to use it in your productions," Dec 2022. [Online]. Available: https://blog.native-instruments.com/what-is-a-sequencer-in-music/

[12] J. M. CHOWNING, “The synthesis of complex audio spectra by means of frequency modulation,” *Computer Music Journal*, vol. 1, no. 2, pp. 46–54, 1977. [Online]. Available: http://www.jstor.org/stable/23320142

[13] T. L. Kunii and M. Inakage, *Frequency Modulation Synthesis*. Springer-Verlag, 1987, p. 381–382.

[14] C. FM, “Sound design basics: Fm synthesis.” [Online]. Available: https://cymatics.fm/blogs/production/fm-synthesis

[15] M. MSP, “Jitter: Cycling ’74max.” [Online]. Available: https://cycling74.com/products/jitter

[16] J.-F. Charles, “A tutorial on spectral sound processing using max/msp and jitter,” *Computer Music Journal*, vol. 32, no. 3, pp. 87–102, 2008. [Online]. Available: http://www.jstor.org/stable/40072649

[17] J. VanderPlas, “Understanding the fft algorithm,” Aug 2013. [Online]. Available: https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/

[18] N. Audio, “Home en.” [Online]. Available: https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

[19] S. W. Smith, *Chapter 8: The Discrete Fourier Transform*. California Technical Pub., 1997.

[20] A. Hayes, “Blockchain facts: What is it, how it works, and how it can be used,” Apr 2023. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp

[21] J. Frankenfield, “Cryptocurrency explained with pros and cons for investment,” May 2023. [Online]. Available: https://www.investopedia.com/terms/c/cryptocurrency.asp

[22] H. Vranken, “Sustainability of bitcoin and blockchains,” *Current Opinion in Environmental Sustainability*, vol. 28, pp. 1–9, 2017, sustainability governance. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877343517300015

[23] BusinessDay, "How does bitcoin work?" Jan 2021. [Online]. Available: https://businessday.ng/news/article/how-does-bitcoin-work/

[24] D. Vujičić, D. Jagodić, and S. Ranić, "Blockchain technology, bitcoin, and ethereum: A brief overview," in *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2018, pp. 1–6.

[25] J. Frankenfield, "Gas (ethereum): How gas fees work on the ethereum blockchain," Jan 2023. [Online]. Available: https://www.investopedia.com/terms/g/gas-ethereum.asp

[26] J. Kastrenakes, "Grimes sold 6 million dollars worth of digital art as nfts," Mar 2021. [Online]. Available: https://www.theverge.com/2021/3/1/22308075/grimes-nft-6-million-sales-nifty-gateway-warnymph

[27] S. Hissong, "Kings of leon will be the first band to release an album as an nft," Mar 2021. [Online]. Available: https://www.rollingstone.com/pro/news/kings-of-leon-when-you-see-yourself-album-nft-crypto-1135192/

[28] Makerpad, "Introducing thirdweb: Can anyone now build their own web3 app?" [Online]. Available: https://makerpad.zapier.com/posts/introducing-thirdweb-can-anyone-now-build-their-own-web3-app