EARFREQ
(Exaggerated Acoustic Reality)
Preliminary Prototype

Matt Adair
s3204584@student.rmit.edu.au

Synopsis -:

The EarFreq Project is chartered with constructing a preliminary prototype that seeks to investigate the hardware requirements for creating object detecting sonar earphones as well as the software that translates these objects into a meaningful and musical soundscape.

The device itself consists of a set of earphones and software that uses ultrahigh frequency scanning of the surrounding area to create a three dimensional map that consists of "regions of interest". These will be limited to moving objects that are within a certain size threshold. These objects are described audibly dependent on where they are located in relation to the user, as well as their velocity and direction of travel.

The occurrence of these objects and how they move is used to create a soundscape of unique quality that is responsive to the particular surroundings of the user. It is intended that this device and its software can be used by people that would enjoy an immersive soundscape, one which is triggered by their environment and could also be used as the basis of research into an audible navigation device.

Description -:

The EarFreq Project consists of a Concept Prototype that is limited to specific research goals and milestones and a Development Prototype that is more open in its aims and configurations. The Concept Prototype requirements are to create a suitable hardware test harness under which software will be developed. This harness will be built upon an existing environment that is well documented and can facilitate rapid prototyping.
The purpose of this prototype is to allow the creation of a software synthesis engine that can suitably respond to a dataset consisting of up to four objects and their associated positions and vectors. Investigation into various issues that research has already uncovered include, but are not limited to, ear fatigue, inter-aural ambiguity, situational awareness and musical styles.

Following the completion of this stage the project then advances to the Development Prototype that will use the Concept Prototype software to investigate the hardware possibilities and limitations that might be encountered. These limitations mainly consist of transducer size, power consumption, range, resolution and tracking speed. The purpose of this prototype is to deliver a test article that is suitable for testing in a street environment.

This test article will be measured against the concept requirement of a device that can respond to the user’s environment of moving objects and create an interesting and dynamic soundscape that evolves over time. It is expected that the earphone device is no larger than an ordinary set of headphones and that the software installed on an Android handset will have enough processor power to meet minimum requirements for object tracking.

Rationale -:

The successful conclusion of this project provides immediate benefits to the developer not just in career progression but in advancing the skills available to manage and construct a device of reasonable complexity. While the construction of the device itself remains within the boundaries of realistic possibility, it does extend the experience of the developer in the fields of research, project management, electronic engineering, software engineering and product development.

The Preliminary Prototype discussed in this document is not intended for retail but instead offers a platform for a product that is marketable to the public in the form of an entertainment application installable on the user's handset. This application would be advertised to potential users as being a set of earphones that create a musical soundscape which responds to the user's immediate surroundings.

Second to this path is the use of the device as a platform for further research into the potential for navigation aids. This research would be required to investigate in greater detail some of the issues already uncovered at this stage, namely inter-aural ambiguity, ear fatigue and situational comprehension. These matters are considered to be beyond the scope of this particular proposal stage and are mentioned here only to highlight their potential.

Ultimately, with the use of personal music devices (iPods et al) and mobile phones that incorporate music players it is easy to assume that most people like to have their own personal soundtrack as they go about their business. The popularity of playlists and commercial music aside, the desire to have music and the need for the emotional connection that music provides is appealing to most people. This device seeks to add to the repertoire of musical instruments available to the average person regardless of their ability to understand musical notation or theory.

With this device, the user is essentially the instrument and one that is played merely by standing on a street or a in a room. As they move or as objects within their area move, new notes are added, evolved and stylistically altered in a musical way. This dynamically changing landscape can offer a multitude of musical events that are unique to a given user, at a given location, at a given time.

Production -:

The Concept Prototype

For this purpose the Eyesweb IDE has been identified as being able to provide a suitable analogy to the conditions expected to be encountered with the final, fully functioning device. Instead of using sonar for object detection perpendicular to the user, this IDE will use camera based, overhead, motion tracking that can generate a given region of interest into an exportable dataset that consists of object location, size and velocity parameters. This dataset will be imported into a synthesizer engine for translation into parameters that can create an evolving soundscape.

Eyesweb allows the developer to rapidly iterate through prototypes that can test various configuration and conceptual changes while providing a consistent dataset for the synthesizer to use. The completion of this stage is measured by successfully tracking four discrete objects through a two dimensional plane and sending a suitable dataset for the Reaktor synthesizer to use in musically representing the object states. An example patch has been created to demonstrate the various aspects and principles required for this research and is discussed in Appendix A. Further to this, the implementation of the prototype is discussed in Appendix B. The translation of object state vectors into musical events and their positional hierarchy relative to the user is discussed at length in Appendix C.

The Development Prototype

This stage consists of two major components, a set of earphones with active and passive ultrahigh frequency transducers and software compiled for the Android operating system necessary to power and control it. This software will include the code necessary to parse the data sent and received by the transducers into usable MIDI controller data that can be sent to the Reaktor synthesiser.

Each earphone contains four transducers that are arranged in a forward, rear and side configuration to allow full coverage of the surrounding area of the user. The active transducers are programmed in such a way as to generate more accurate and rapid tracking of any objects in front of the user. Whereas the side and rear transducers are programmed to offer minimal distractions to the user that would ordinarily be concentrating on the events taking place in front of them.

One of the major tasks of this stage is to investigate the realities of using components whose specifications are fixed and finite. For instance, the transducers have an effective range of approximately ten metres and a relatively high power drain when in active mode. Another major aspect to research is the ability of the hardware to provide a suitable dataset for the software to parse into a musical soundscape comparable to the one achieved with the Concept Prototype.

The Production Process

The aim of this project is to produce a Proof of Concept prototype test article that is suitable for use in a street environment, using earphones and software compiled and installed on an Android 2.3.4 handset that has a working USB host port and headphone socket. See Appendix D for a discussion of the software development process.

The first stage is to develop the Musical Principles necessary to create a dynamic soundscape based upon object-state vectors by using a motion-capture IDE . Once this has been demonstrated, the software synthesiser is then used as the basis for testing the sonar hardware and any software necessary to control it. The intention here is to replicate the same quality dataset achieved via motion-capture in an aerial perspective using the sonar transducers in the horizontal plane.

This is the rationale behind separating the project into the two distinct prototypes of Concept and Development as there are two distinct components involved, namely hardware and software, that will present their own unique challenges. It should be noted that this development needs to take place in order and it is for this reason that attempts have been made to utilise existing software environments to lessen the development times.

From reading the appendices it is possible to ascertain the level of research necessary for any given stage of development to advance the project closer to a Test Article suitable for use in a street environment. The production of this article for use in research into its function is considered to be the ultimate aim of this project.

Timeline -:

The Schedule Description

The project has five major and twenty-six minor milestones to measure performance and development against. These milestones cover the construction of the Test Harness rig, Musical Principles envelope expansion tests, volunteer testing, breadboard prototyping of hardware, fabrication of the Test Article and software development for final acceptance testing and delivery of a compiled runtime. The project schedule is expected to take place within a forty-seven week period with several contingency weeks situated prior to major milestone achievements.

Milestone	Project Activities	Time
1.0	Eyesweb Test Harness construction and function assessment	Week 1
	Research alternative methods of object tracking within patch	Week 2
	Patch-lock, integration and testing with synth application	Week 3
1.1	Record film of object movement tracks with variations of number objects, velocities, etc	Week 4
1.2	First object oscillator envelope expansion tests	Week 5
1.3	Second object oscillator envelope expansion tests	Week 6
1.4	Third object oscillator envelope expansion tests	Week 7
1.5	Fourth object oscillator envelope expansion tests	Week 8
1.6	Integration tests, oscillator envelope lock	Week 9
2.0	Musical Principles envelope expansion tests	Week 10 – 11
2.1	Prepare Test Harness with candidate music styles as switchable presets as per tests wk 10-11	Week 12
2.2	Volunteer testing of musical styles presets	Week 13
2.3	Assessment 1 of Musical Principles generated music style presets, incorporate changes	Week 14
2.4	Secondary testing of Musical Principles if required or contingency period for other tasks necessary	Week 15
2.5	Assessment 2 of Musical Principles and current state of project development.	Week 16
	Contingency Period 1	Week 17 - 19
3.0	Test Article hardware testing and configuration	Week 20
3.1	Construct and test of single sonar transducer breadboard prototype	Week 21
3.2	Construct and test complete Test Article breadboard prototype	Week 22
3.3	Circuit and software integrity checks for breadboard prototype	Week 23

3.4	Breadboard prototype envelope expansion tests	Week 24
	Contingency Period 2	Week 25
4.0	Construction of Test Article, test and comparison to breadboard prototype envelope	Week 26 -27
4.1	Test Article and Test Harness integration tests	Week 28
4.2	Replication of milestone 1.1 test conditions in Test Room Facility	Week 29 - 30
4.3	Assessment 3 of Test Article performance	Week 31
4.4	Limited street based testing and refinement of Test Article and Test Harness	Week 32 - 33
	Contingency Period 3	Week 34
5.0	Android software development preparation	Week 35
5.1	Implementation 1 of hardware control, synth module, logic and interface layers	Week 36 - 40
5.2	Assessment 4 and Contingency 4	Week 41 - 42
5.3	Implementation 2 and final integration of hardware and software components	Week 43
5.4	Assessment 5 in Test Room Facility	Week 44
5.5	Replication of milestone 4.4 test conditions	Week 45 – 46
5.6	Acceptance Testing and Delivery	Week 47

(Fig. 1: Table - Project development schedule)

The project development table above highlights key areas and the expected time to complete before the next stage can be started. Each time period is assumed to consist of weekly blocks and it is assumed that within that period other tasks will be completed as needed, such as management and review processes. There is also a variance in the workload that a given project activity would require for completion and it is assumed that within these periods any other previous tasks can be completed or advanced tasks started earlier.

The two major workload tasks identified are the construction of the sonar device and the main software implementation in Android. These two have ample allotted time for completion as well as a number of contingency periods should the need arise. The only other major milestone that has some ambiguity attached to it is the research and programming of the Musical Principles. It is assumed that by the end of Week 19 any development in this area will have to be ceased regardless of the state. In this instance it is necessary that the project does not get delayed by endless iterations of musical styles but instead can progress with a number of presets that are reasonably unique to one another and allow a working model to be installed in the Test Article.

In the table above, “envelope expansion” tests refer to a process of determining the number of parameters that can be changed, the extent of any parameter changes, visibility of parameters to the user and any interface elements necessary. Together with the actual parameter conditions it is also necessary to measure and record the results of any parameter changes as well as determine default or preset states.

Research -:

To determine the scope of this project it was necessary to conduct research into existing methodologies of using sonar devices in such areas as robotics and travel aids for the blind. Several important articles were published that examined the theory behind sonar use and how humans can respond to any information gained from them. The first major article (Bujacz, et al, 2010) goes into great detail about research being conducted at the Technical University of Lodz, Poland into developing and trialling such devices.

One of the most important aspects noted here is the theory behind head related transfer functions (HRTFs), or understanding the way the human ear functions as well as the way the human mind processes information received by the ear. This research is based upon an earlier work (Pec, et al, 2008). The theory is complex and beyond the scope of this project at its current stage, suffice it to say that any intentions of applying this device to a navigational aid would change its outcome and remove any musicality from the tones generated.

From limiting the device to being one that merely responds to events around the user by detecting an object using sonar and then tracking its vector through the user's local space, we then move to a working implementation of a sonar device. A similar scale and technology is evidenced in the device called a 'K' SONAR which is a hand held device attached to a guide cane that emits ultra high frequency tones for use in object detection. It can measure not only the direction of an object but its distance and this information is transmitted to the user as a series of clicks and chirps via an earpiece (Kay, 2006).

The 'K' Sonar device utilises off the shelf transducers that can be sourced in cheap quantities and at various specifications. One of the most common transducers, found in robotics and industry, is the LV-MaxSonar which offers a good range of specifications (MaxBotix, 2011) to meet the requirements of this project. The theory and practice of building a simple prototyping circuit for these transducers is well documented (Eady, 2008) with circuit diagrams, schematics and sample code freely available from many sources on the internet ranging from hobbyists to peer-reviewed articles. It is because of this wide use that it is with a high amount of confidence that these transducer circuits can be successfully employed in the Test Article.

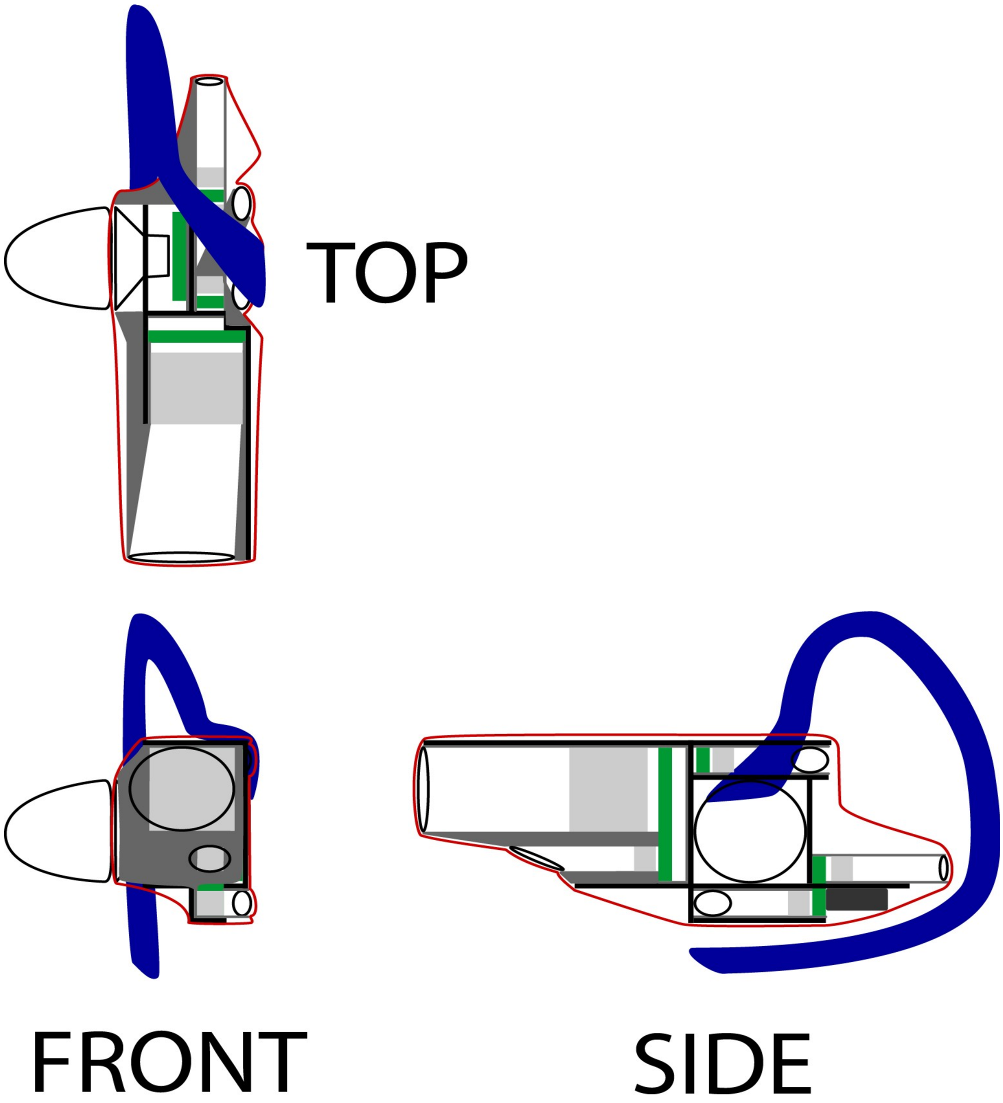
A basic understanding on how the human mind processes audio events is of importance in forming a basic framework for the synthesis the project will be using. How an object's location and vector can be translated into a meaningful and memorable musical event is the basis of the first twenty weeks of the project schedule. Research has been conducted (Morland, 2008) into the types of audio events that can be readily distinguished form one another and the kinds of information subconsciously attached to these events by a user. While this research has uncovered the fundamentals of echolocation, it is the intention of this project to extend these principles into more musical results that aren't limited to providing the user with exacting spatial co-ordinates of objects to allow navigation.

Gaining a good and usable dataset from the sonar device forms the second stage of the project, the Development Prototype. Prior to this is the research into the Musical Principles that form the main component of the Concept Prototype. At this stage of the project it is more important to get a functioning device constructed before examining at any great length the possible musical styles producible by the device. It is assumed that research will be conducted into musical theory, psychoacoustics, semiotics and other fields necessary to build up an understanding on how long term use of the device would affect the user. For now, it is enough to get a working device that constructs a soundscape based on any object state vectors surrounding the user and in order for this to be demonstrated a number of preset musical styles limited to electronic synthesis will be used.

Designs -:

The most visible component of the project are the pair of earphones used to transmit and receive ultrahigh frequencies in all directions around the user as well as the speakers to play back the resultant soundscape. The main limitation in the design of the earphones are the sizes of the transducers, their associated circuits and any dowels or baffles required to ensure a clean signal is transmitted and received by the hardware in the required direction. It is assumed that at the very least, the forward facing transducer pair operating at greatest resolution will need to be the most accurately constructed to operate at nominal specifications. The side and rear facing transducers are free to be manipulated as to their size and dowel reach to within reasonable limits of scale.

(Fig. 2: Earphone hardware – concept version 1)



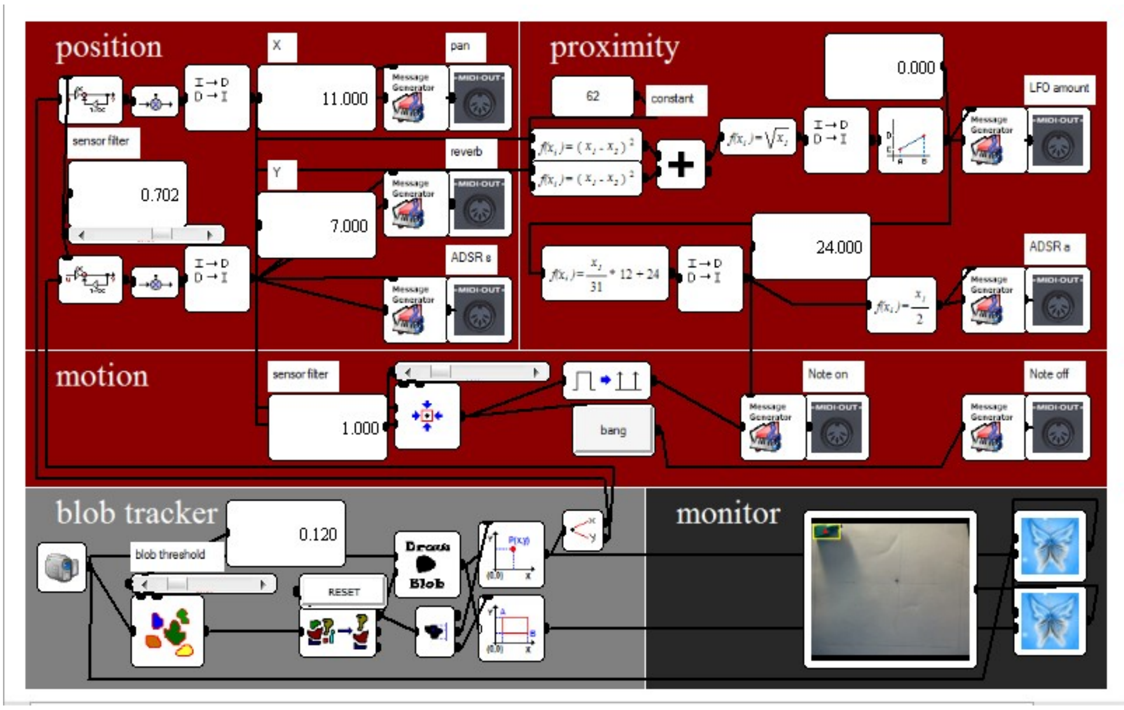
(Fig. 3: Earphone internal hardware – concept version 2)

The current designs for the hardware component are based upon over-the-ear earphones that have popular and common Bluetooth earphones as their precedent. These designs are assumed to have sufficient housing space for all the required component and would utilise a two-way cable running to the handset that would carry both sonar data and audio separated by shielding. If at any stage the total weight and/or size of the device goes beyond what is deemed comfortable to be supported and worn over-the-ear then the designs will shift their requirements to full sized headphones that enclose the ear.

The requirements for the software component are that the application is coded in Java and compiled to the minimum target of the Android operating system version 2.3.4. It should have sufficient processor power to control and interpret any data received by the transducers, generate a soundscape governed by the Musical Principles presets and offer limited control to the user. This control will be presented to the user as a primary interface screen with controls for overall volume, stereo spread, amount of sonar activity, rate of change and other musical parameters deemed appropriate.

(Fig. 4: Handset interface simulator – concept version 1)

Appendix A -:



Motion-capture demonstration
(Fig. 5: image: Eyesweb patch BlobExtractReaktor03.eywx – Blobtrack-position1.jpg)

In this instance the video camera and motion capture section of the patch are contained within the light gray area labeled “blob tracker”. A blob is an associated group of same colour pixels that occur within a given radius of each other. Once a blob has been identified it can be tracked as it moves all the while generating barycenter and bounding box Cartesian co-ordinates. Its these co-ordinates that are then drawn onto a monitor window as well as being sent to the logic parts of the patch that are located within the red areas.

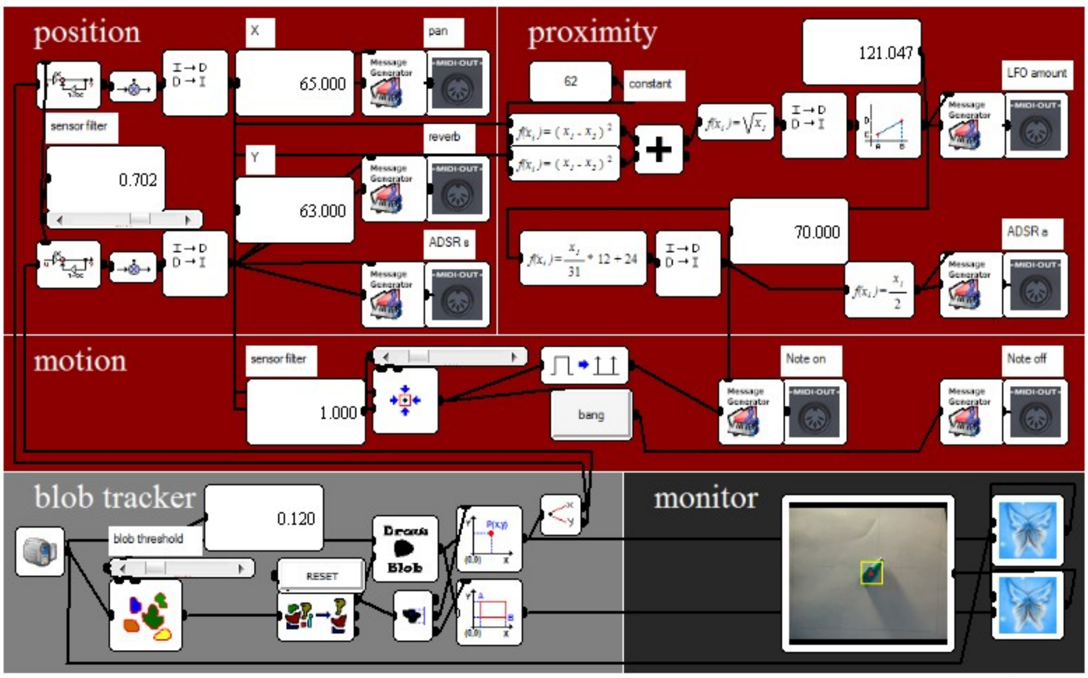
The first logic area is labeled “position” and its here that basic X and Y positional data is sanitised and then converted to usable MIDI data within a translated range of 0 to 127. This data is then sent to MIDI out ports with X going to Pan position and Y going to both the Reverb and ADSR sustain ports. This data is also sent to the other two logic areas that are labeled “proximity” and “motion”.

The “proximity” area evaluates the function (((x - a)^2) + ((y - b)^2)) = r^2 where a and b are the co-ordinate centres, in this case a constant value of 62. This function determines the proximity of the blob to the centre of the viewing area which is representative of the users location. This data is rescaled to usable MIDI data and then sent to the LFO port as well as a second function to generate note pitch and ADSR attack information.

The note pitch function takes the proximity generated information and converts it to data that is within the range of two octaves. This is then sent to a further function that halves its values before sending it to the ADSR attack port, as well as then sending it to the “motion” section. This section has a basic function that checks whether the blob has moved on either the X and/or Y axis and if it has then it sends a Note On and an All Notes Off trigger to their respective ports.

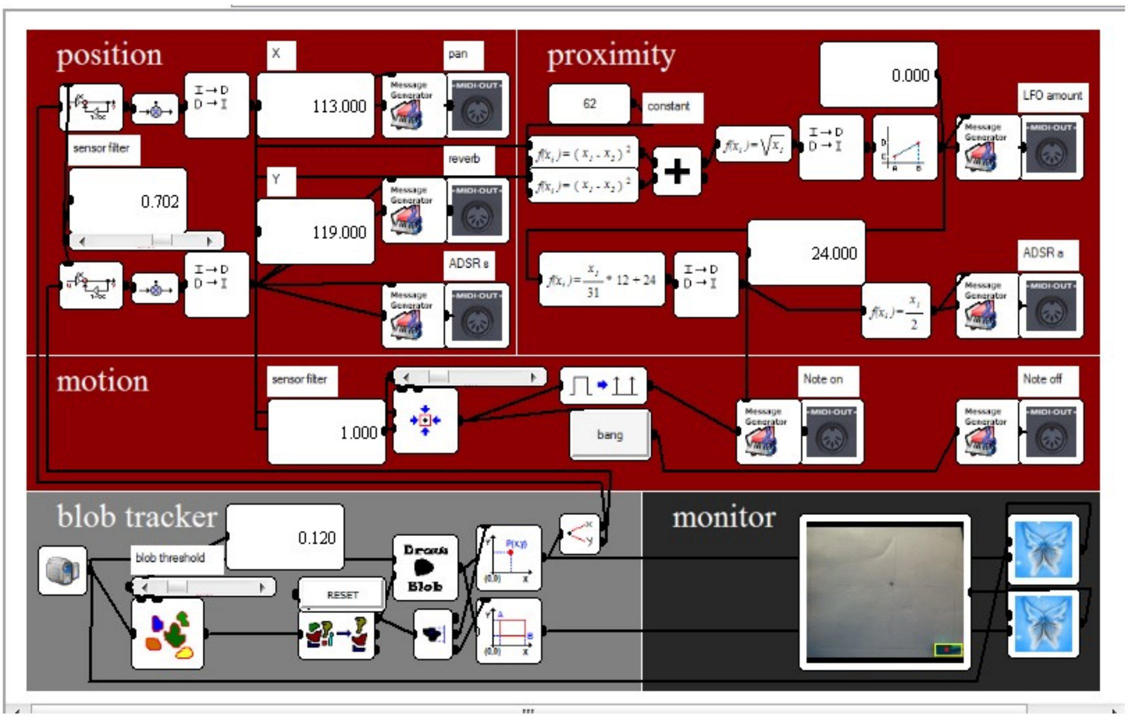
The data acquired by this particular patch is designed to generated usable MIDI control parameter changes that result in distinct audio events for the purpose of researching the tonal qualities required to construct a musically interesting soundscape that is directly derived from the users immediate surrounds. Other controls visible in the patch and not mentioned here are related to successful acquisition and tracking of a given object into stable MIDI data.

In the demo example (*wmv: BlobExtractReaktor03-demo01*) a green object is tracked throughout the viewable area with its yellow bounding rectangle and red barycenter discernible. In this patch, the monitor window shows a bird's-eye view of the area that surrounds the user with their location at the centre. The top of this view represents the area in front of the user, the bottom represents the area behind the user and left and right are relative. When a blob is acquired in the top left *(Fig. 5)* the resultant audio is low in pitch, short in attack and sustain, zero LFO and zero reverb.



(Fig. 6: image: Eyesweb patch BlobExtractReaktor03.eywx – Blobtrack-position2.jpg)

As the blob is tracked across the view to the centre *(Fig. 6)* then the audio quality changes to being high in pitch, high in LFO and middle values for all other parameters.



(Fig. 7: image: Eyesweb patch BlobExtractReaktor03.eywx – Blobtrack-position3.jpg)

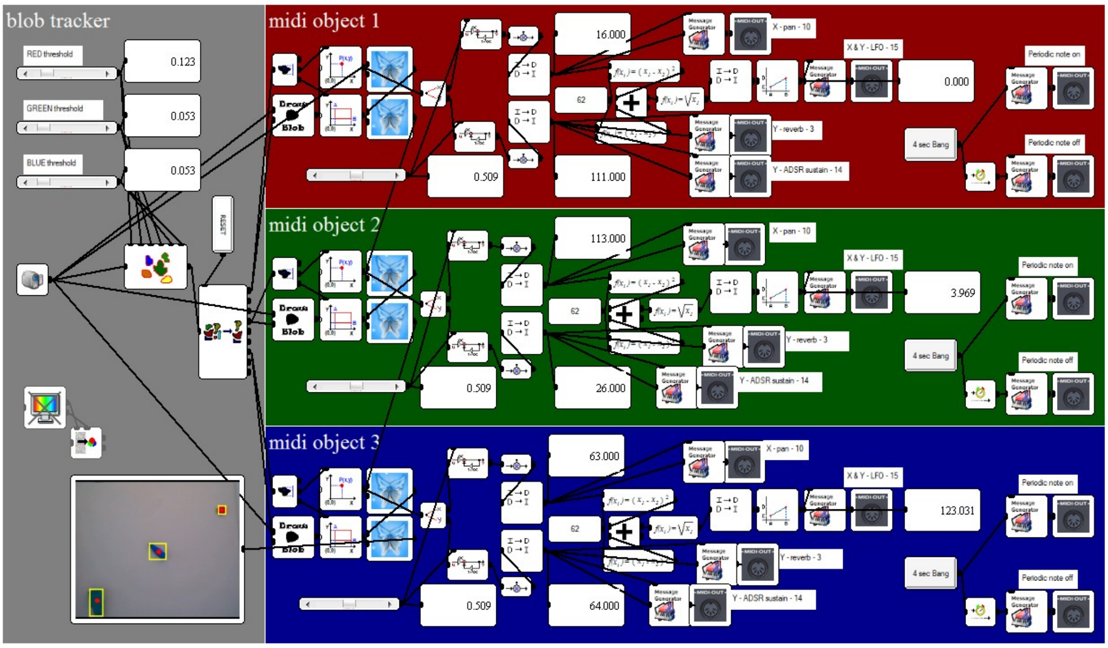
By the time the blob is tracked to the bottom right *(Fig. 7)* then the pitch and LFO revert back to being low and the attack, sustain and reverb move to high.

This patch suitably demonstrates that usable MIDI data can be generated in a testing environment so that further research can be undertaken into the types of synthetic tones, note, chord progressions and effects that can be utilised to generate a soundscape generated by vector objects.

Appendix B -:

Test Harness utilisation

The Concept Prototype is specifically chartered with the task of utilising a suitable test harness for the development of meaningful synthesised sounds and the principles of their creation. The software used to create these sounds (Reaktor) offers the ability to create generic synthesis modules that can be replicated in the final software compile target of the Android operating system.



(Fig. 8: image: Eyesweb patch BlobExtractReaktor02.eywx – BlobTrackerReaktor02.jpg)

This Test Harness needs to be able to generate MIDI data limited to four channels for the synthesiser to provide a voice for each as well as the necessary number of control change parameters to allow manipulation of the audio. In the example patch (Fig. 8) three separate objects are tracked, represented by the colours red, green and blue.

The Test Harness will consist of the following components:
Windows 7 laptop computer
Eyesweb 5.2.1
Reaktor 5.1.1
800 pixel x 600 pixel webcam
4 coloured pucks for tracking

Initially the software will be configured to track a single object to allow the synthesiser module to be programmed to translate the object's vector state into musical events. Once this is achieved then the other three modules can be turned on and programmed to track three other objects simultaneously. Balancing the full complement of audio events from four objects into a usable soundscape is expected to be quite arbitrary in terms of musical tastes so it is expected that further scope for user interaction with the software via controls will be incorporated.

(Fig. 9: image: Reaktor patch ReaktorSynth01.ens – BlobExtractTrackerReaktor03-synth.jpg)

A demonstration patch (Fig. 9) was constructed using the Reaktor software and shows Object1 consisting of a Tri/Saw wave oscillator that is manipulated via various controllers and has its audio signal sent to the Object1 reverb module where the signal is outputted with its dry/wet balance controlled by the object tracker.

Research into the principles of creating meaningful soundscape elements will conclude once a reasonable ability to track and discern the differing musical elements is achieved. The state of the software at this point can be locked and the project can then advance to the Development Prototype which replaces the motion capture software with sonar hardware and the necessary software that can provide similar datasets based upon line of sight objects that vector along the same plane as the user.

Appendix C -:
Software Synthesis principles

One of the most fundamental aspects of this project is to be able to translate various things around the user, referred to as object state vectors, into an interesting soundscape, sometimes referred to as musical events. Several limitations are placed upon this with the major one being the scope of this soundscape being constrained by the ability of the human ear and brain to comprehend to a reasonable level only four discrete audio events at once.

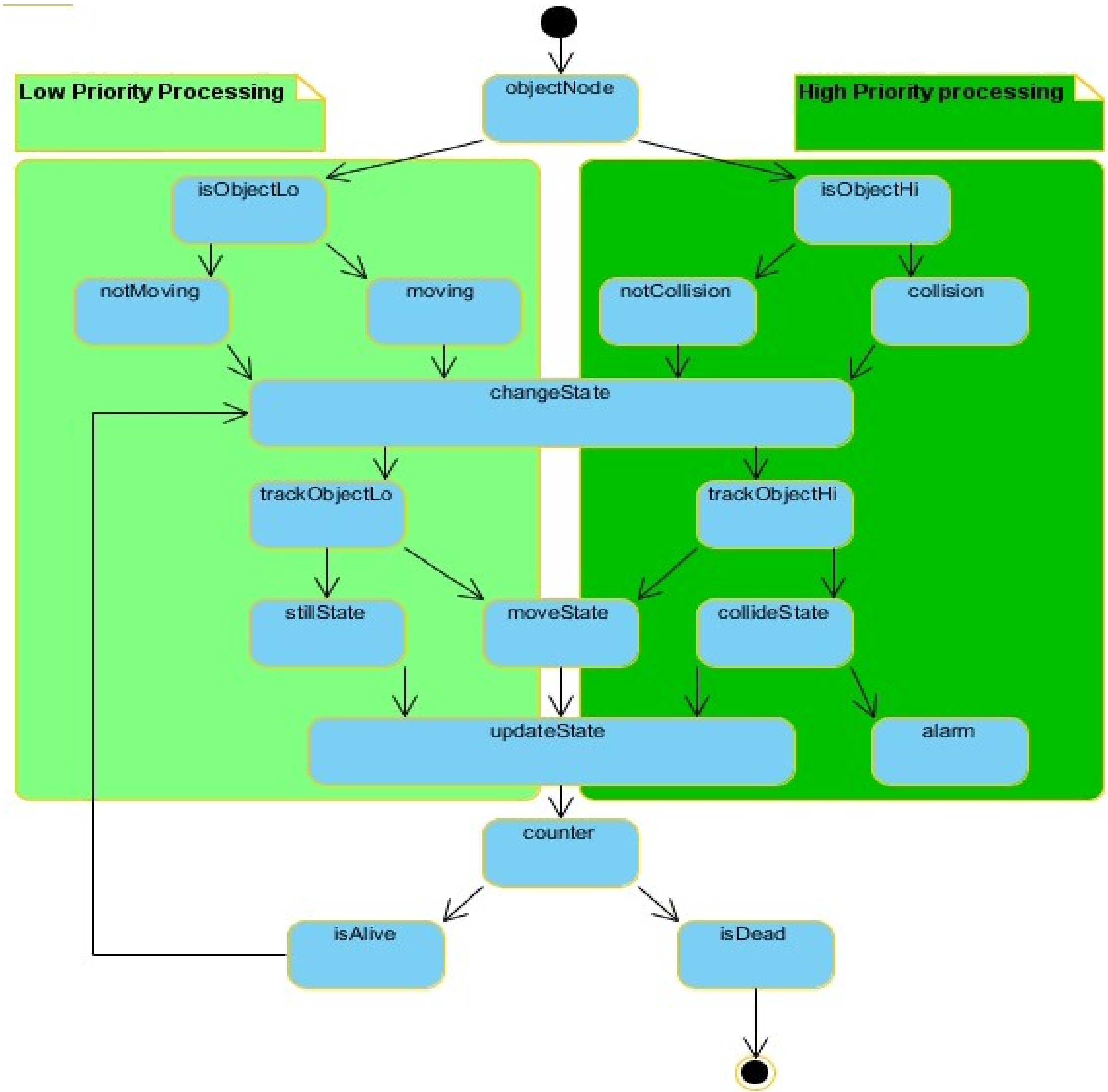
It is then necessary to develop a process of organising any object state vectors detected by the device into a meaningful hierarchy with proximity and direction of travel relative to the user as being the topmost. Along with this order of priority it is necessary to develop a consistent and efficient theory for the translation of object states into a potentially complex soundscape.

(Fig. 10: image: concept diagram – EarFreq-rev2.0.jpg)

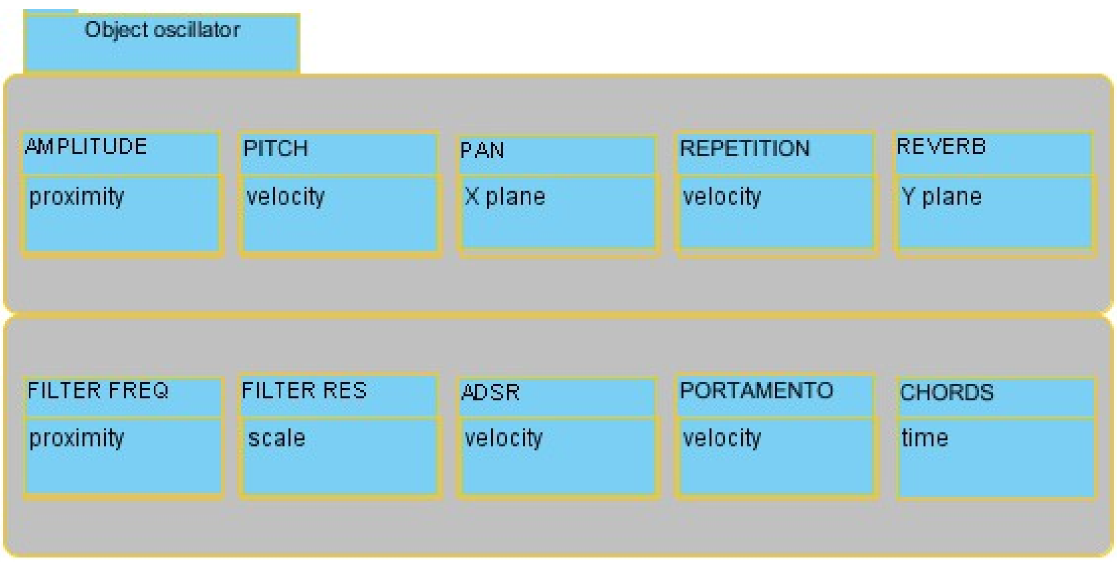
The above diagram (Fig. 10) shows a second revision of the EarFreq concept with its component parts and their descriptions. This is then further broken down into the software state machine diagram (Fig. 11) that examines the major functions and theory behind organising the tracking of objects based upon priority and hardware processor resource allocation.

An example of the theory for the musical state rules is shown (Fig. 12) with a single object's oscillator represented with its initial controllers. These controllers respond to various object state parameters which are also shown in the diagram as being dataset components that translate into musical terminologies.

Included in the media folder for this project proposal document is an file called SoundscapeDemo.mp3 that is an example of translating X and Y co-ordinates of several objects including three people, two roads and a doorway as they move relative to the user. This demonstration does not utilise the phase shifting process that allows more accurate placement within 3D space but instead relies upon simple stereophonic effects.



(Fig. 11: image: state machine diagram – StateMachineDiagram1.jpg)



(Fig. 12: image: musical state rules diagram – MusicalStateRules.jpg)

Appendix D -:
Software Development process

The project will adopt a systematic and organized approach using the appropriate tools and software engineering techniques to achieve a cost effective and productive development process. This involves the use of an Object Oriented methodology within an evolutionary prototyping process and constructing formal processes for testing and problem resolution.

The analysis process within the Object-oriented methodology involves gathering a requirements specification – a description of what the system is expected to provide, and identifying concepts, classes and objects from this. The design phase is the development of a conceptual model from these concepts, classes and objects. The project will be using the Unified Modeling Language (UML) to construct a conceptual model. The implementation phase involves the programming and testing of the system, which begins when the conceptual model has been developed into a detailed design.

An evolutionary prototyping process cycles through several designs incrementally improving upon the final product with each pass. This process is performed in a Rapid Application Development (RAD) environment where the emphasis is on short cycles of analysis, design and implementation, with the focus being on the production and evaluation of a working system.

The reasons for selecting an evolutionary prototyping process are:

- Requirements are well defined.
- Quickly evaluate and refine interface and architectural design.
- Address and resolve any issues before they become entrenched problems (effective risk assessment and reduction).
- Use of high-level languages within Integrated Development Environments (IDE) such as Eclipse, Eyesweb and Reaktor enable fast and efficient implementation.

Constructing a formal testing process will involve designing test cases for each stage of testing as required. The testing stages to be addressed are:

- Unit Testing – testing of individual functions and routines.
- Module Testing – testing of objects and classes.
- Integration Testing – testing modules integrated into sub-system components.
- Acceptance Testing – user testing of the system as a whole.

Constructing a formal process of problem resolution will involve recording the details of any issues [errors, problems, bugs] found, determining who is responsible for resolving the issue and allocating priority. The status of the issue will then be updated when any changes or resolution of issue is found. The project will also need to perform periodic audits of the status of all unresolved issues.

Bibliography -:

Bujacz, M., Pec, M., Skulimowski, P., Strumillo, P. (2010). *InTech-Sonification of 3D Scenes in an Electronic Travel Aid for the Blind*. Politechnika Lodzka, Poland

Pec, M., Bujacz, M., Strumillo, P., Materka, A. (2008). Individual HRTF Measurements for Accurate Obstacle Sonification in an Electronic Travel Aid for the Blind. *International Conference on Signals and Electronic Systems*, Krakow, Poland, September 14-17, 2008

Kay, L. (2006). *'K' Sonar The Handbook*. Bay Advanced Technologies Ltd, Auckland, New Zealand

MaxBotix Inc. (2005). *MB1260-MB1360 Datasheet*. PD10019b, Brainerd, United States

Eady, F., (2008). *Building a Sonar System.* Servo Magazine, June, 2008, Nth Hollywood, United States

Morland, C., Mountain, D. (2008). Design of a Sonar System for Visually Impaired Humans. *Proc. of the 14th International Conference on Auditory Display*, Paris, France, June 24-27, 2008