M.O. Abbott
MUS 499B
Spring 2014
Prof. Sever Tipei

MUS 499B Final Report

This semester in MUS 499B, "Project-Oriented Computer Music", I made considerable headway in a project that has been percolating in the back of my mind for years. My idea is to make a sonification system for a Western astrology natal chart, and to write a computer program which would automatically generate a complete electronic piece for the user, taking nothing more than the user's birth time and place as input. The sonification system would reflect the traditional metaphysical meaning of the astrological planets, signs, and aspects, and other significant characteristics of the Western birth chart. Additionally my scheme would audibly represent the actual orientation of the planets and their angular relationships, from the perspective of the user's birth time and place. Ideally the user could use the program over the internet, and have their piece e-mailed to them or posted for download. Long-term, the entire process should be automated, so that the everything is handled by computers and there is no manual interaction required on my part once the program is in place.

Over the course of the semester it became apparent that this will be a long-term project, with multiple distinct technical challenges to be surmounted. My programming abilities, while mostly adequate for use in pre-existing musical coding environments such as Grace or SuperCollider, require considerable bolstering before I would be able to write an application from the ground up that does everything I want to do for this project. That said, during this semester I had much success in addressing the musical side of the problem – I established an astrological sonification system that meets my criteria as stated above, and I produced test sounds using data from my own birth chart, allowing for a preview of the results.

My initial step was to ascertain which data I wanted to use in the piece, and to either acquire or calculate that data. Fortunately there are many free websites that generate birth chart information, and at www.astro.com I was able to get this information in a text format that is easy to extract into a coding environment. From www.astro.com I retrieved sign, degree, velocity (speed and direction of a planet's apparent motion around the Sun), latitude (planet's position with respect to the "ecliptic" – the plane between the Earth and Sun), and length (distance from Earth) for each planet used in a typical Western chart. However, I did not feel this was enough information on which to base a piece, so I also sought astronomical data from other sources. The best resource I found for astronomical data is a freeware program called Xephem, which offers a wealth of astronomical data, some of which is astrologically useful and most of which is not. Specifically, the data I used from Xephem includes magnitude (brightness), PMDec (vertical velocity, with respect to the ecliptic), size, and phase. After combining astrological and astronomical data from these two sources, I presently have 9 distinct values for each planet. In the long view it would be ideal to have my program simply calculate all of the data I retrieved from www.astro.com and Xephem, but as stated below, that is beyond my current abilities.

My next problem was to develop a rational by which to craft an effective musical interpretation of the astrological birth chart. An astrological chart is basically a big circle, broken into 360 degrees. The traditional "starting point" of this circle is the Ascendant, which is the point in the sky that is ascending over the Eastern horizon at the time of birth. Astrological houses (of which there are 12) begin at this point and go all the way around the chart counterclockwise, such that the 1st and 12th houses straddle the Ascendant. As such, I chose to interpret the Ascendant as the beginning and end of the piece. The piece follows the houses in a counterclockwise fashion around the chart, and musical events are triggered whenever a planet or important astrological point is reached. The time scale of the piece is to be determined, or perhaps could be defined by the user.

The astrological planets (which traditionally include the Sun, Moon, Pluto, and Chiron) are scattered about the edge of the circle described above. Exact placement of the planets, and the exact values informing each planet's statistics, will depend on the user's birth time and place. The most significant points in the chart are obviously the planets themselves, and major musical events will certainly occur when a planet is reached as the piece works around the circle. Each pair of planets also has two "midpoints" – these points are halfway between a two planets along the edge of the circle, and are traditionally thought to possess a meaning that is a combination of the two planetary energies. Per my sonification, each midpoint will also receive a musical treatment. With planets and their midpoints, there are over 300 separate events in a given run of the piece. The ordering and timing of these events, and therefore the density, will be different from chart to chart – in my own chart the distribution is mostly even, with a few spots where events bunch together more, and some areas with nothing whatsoever. Midpoint calculations were performed via manipulations of sign and degree values for all planets, for which I used Grace.

Most of the actual data sonification occurs in the design of planetary synthesizers. Each planet has its own synth; I have built these synths using SuperCollider. In astrology every planet "rules" one or two signs. Every sign has a "cardinality" and an "element" (these are innate to the planets, regardless of chart placement). To determine the attributes of a planetary synth, the cardinality and element of the signs that the planet rules are considered, along with the cardinality and element of the sign in which the planet actually resides in the chart. Hence, some characteristics of the planetary synths are consistent in every realization of the piece, whereas the planet's chart placement will define additional characteristics of that synth.

The actual statistics for each planet, as obtained from www.astro.com and Xephem, are then mapped to certain parameters of sound used in the corresponding synth. Below is the exact breakdown of the parameter mapping:

degree = default frequency
velocity = panning modulation frequency
latitude = panning center location
length = reverb "wetness"
magnitude = amplitude
PMDec = frequency modulation
size = duration
phase = reverb "size"

The exact numerical values of the input parameters had to be rescaled in most cases, to a range of values that are appropriate for use in SuperCollider. For example, the "degrees" input values range from 0-30; this range was scaled to 20-1250 for SuperCollider to represent hertz values for frequency. I performed all such rescaling and data manipulation using Grace.

I believe my mapping scheme handles the parameters fairly well in terms of astrological meaning and astronomical reality. The most quizzical for me has been "phase", which I do not feel corresponds well to any musical parameter I have tried as yet. Some parameters, for example amplitude envelopes, have admittedly not been dealt with yet, and could perhaps be handled by reusing existing values. I also believe that velocity might be better represented by a panning envelope rather than a constant modulating frequency – the exact application of such a panning will depend on how many channels I am using for playback, which is to be determined. The layout of an astrological chart does suggest front/back orientation, and as such a 4+ channel sound system would be a better representation than simple stereo.

In terms of crafting musical events, the plan is straightforward – when a planet is reached, the synth for that planet is played, using the planet's default frequency (derived from the planet's degree). For midpoints, which are combinations of planetary pairs, both planetary synths for the given midpoint are played simultaneously. Both synths use the same frequency, which is the average of both synth's default frequencies. The amplitude is also significantly lowered, since midpoints are much less important than the planets themselves.

However, there is more to event construction than playing one or two synths – in fact, several more synths may (and usually do) play a role in a single event. The method for constructing more complicated events is based around the idea of astrological "aspects", which play a significant role in chart interpretation. As stated above, the astrological chart is basically a big circle, with points scattered about its edge. Aspects are determined by measuring a point's distance (usually in degrees) from another point along the outside edge of the chart. When this distance is equal to, or approximates, a low-integer division of the 360-degree circle, the two points are said to be "in aspect", and the distance between the points determines what type of aspect is formed and interpreted. For example, if two points are about 180 degrees from eacher, 360/180 = 2, and this is called an "opposition". If two points are about 120 degrees from each other, 360/120 = 3, and this called a "trine". In astrological interpretation it is rare to go beyond 12 divisions of the circle – as such 12 is the furthest I go in this sonification, although theoretically it could go much further. As these relationships are almost always approximate, rather than exact, the allowable margin of error can also be adjusted by the programmer or the user. In this case, all aspects for all points were calculated using Grace, and I used margins of error that are typical for Western astrology chart interpretation.

Musical treatment of the aspects is as follows: When a point is reached, the planetary synth(s) at that point is triggered (at its default frequency). That frequency is then treated as a fundamental for points that are in aspect to the point reached. Any aspected points (including midpoints) are given frequency values that are overtones of that fundamental – the synth(s) of an aspected point is played at a frequency equal to the integer division of the circle formed by the aspect, multiplied by the fundamental. Additionally, the amplitude is lessened in each successive overtone, mimicking the typical spectral pattern seen in harmonic instruments. Using this method, each event becomes harmonic tower of planetary synths – when considering midpoints, a single event can easily use 30+ synths to represent all aspected points. (That figure can be increased or decreased by adjusting the margin of error used to qualify aspects.) Any given realization of the piece will include over 300 of these multi-component events, and while the default frequencies are static, the planetary synths will more often be played as an overtone of some other synth or synth pair.

Looking ahead, there is much more work to do on this project, and this is likely to demand attention for some time. A shorter-term goal is to make a working mockup, which will accept input information manually and generate a piece. I can write script using Grace that would accept all of the input values and generate code that could be quickly played back in SuperCollider. At that point I could quickly make additional realizations of the piece from other birth charts. Once I have a set of sample runs I might wish to refine my sonification system, having heard more possible results. If my goal is to totally automate, it would be preferable to avoid using Grace or SuperCollider entirely, and to build my own lightweight audio engine that would handle all of my needs more efficiently. Following that, it will be necessary to write code to automatically calculate all of the birth chart and astronomical values, without having to manually refer to secondary sources as I do now. Fortunately, much of the mathematical information needed to perform these calculations is freely available on the internet. However, to do this I would need to become proficient in a free-form programming language such as C++, which is certainly a long-term goal. Beyond that there is the technical problem of making the whole thing work over the internet, which may require use of Javascript, and host of other technical and logistical decisions that will be largely based on how my application is developed leading up to that point.

In conclusion, I am happy to have made significant progress on this project this semester. The input of my instructor and peers in the course was particularly helpful; I received excellent ideas with respect to sonification and a laundry list of technical issues that I will need to consider as I move forward to make this project a reality. I have clear ideas about what I need to do next, and the course allowed considerable insight into music programming approaches.