# 2 - Literature Review

## 2.1 - Generative Algorithms

Generative algorithms appear in many forms, but at their core they work by taking the current step and then based on this pseudo-randomly generating the next step. How the algorithm generates this next step can be based on pre-composed music, which the program analyses and uses to work out the musical structure. It can also use a rule-based system, created from music theory, to generate this structure or a combination of these in a hybrid system. In the past decades, there have been a number of variants that have been utilised to produce in-game music, some of which are more appropriate than others. This section will outline some examples of these and their uses.

### 2.1.1 - Neural Networks

Neural networks were originally inspired by biological networks (Bishop, 1996), such as the human brain and the way in which it processes information with a series of interconnected processing units. It can also modify the connection between these units, thus allowing it to learn (van Gerven, 2017). Neural networks have long been used for pattern recognition, for example, if a human analysed a picture of letters it would be able to learn to identify or produce images it had not seen before.

This suggests that neural networks would be an appropriate algorithm for algorithmic music production. For example, Colombo et al. (2017) developed a system which produced music in the style of Irish and Klezmer folk, as this is the style they chose to train it on. Johnson (2017) intentionally used pieces from multiple genres so the trained network produced more ‘rounded’ music. However, in both of these experiments their chosen analysis method was to look at the distribution of notes compared to the training music. While they found that the produced music was mathematically similar they did not get human experts to evaluate how ‘good’ the music sounded, a common trend in many papers ADD MORE HERE. This is not always the case, for example, Prisco et al. (2017) utilised an algorithmic evaluation and a group of musical experts, who all had more than ten years in the music field. They found the music produced was of high quality and stylistically coherent, the experts were also able to point out flaws in the rhythmic elements of the music, giving Prisco et al. avenues for future work. However, as many of these papers’ goals are not focused on producing the music in real-time they make no mention of the time it took for their systems to be trained, produce the music, or their computational cost. This would make their use in a video-game scenario questionable.

### 2.1.2 - Genetic Algorithms

Like neural networks, genetic algorithms are also inspired by nature, however these are based on evolution. This works by generating several options, the one which best fits a specific set of criteria is chosen and the more options are generated based upon this. This can then be repeated until a satisfactory outcome is reached.

An example of this was a system created by Ostermann, Vatolkin, and Rudolph (2017) which generates a drum beat, in real-time, based on how a live band are performing. This generated a MIDI file of the players’ output and used this to choose the drum beat which fit best with the current musical solo. The musicians reported that this system was mostly successful when improvising along with a band. However this would then require some sort of pre-generated music to be present before the drum tracks could be altered, whether this was pre-composed or procedurally generated. The purpose of this system was as a practice aid for the musicians, rather than to create finished pieces.

Another way genetic algorithms have been used is to generate fitness functions, which can then numerically evaluate musical pieces, this technique was utilised by Loughran and O’Neill (2017). This would then allow for the music and the fitness functions to ‘evolve’ together. However the authors state that the music produces were not overly impressive, coupled with the fact that these generation techniques required many melodies to be generated at a time. This procedure would drastically increase the amount of processing power given over to the music production, which could have an impact on the performance of a game as a whole.

### 2.1.3 - Markov Chains

Markov chains are used for modelling a finite number of states, in terms of music this would be the notes in a given melody, and the probability of moving between them (Snodgrass and Ontañón, 2014). For example, if a small musical melody is ‘D, D, F#, D, D, D, E’, the frequency distribution matrix would be as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Next Note | | | Total Frequency |
| D | F# | E |
| Current Note | D | 3 | 1 | 1 | 5 |
| F# | 1 | 0 | 0 | 1 |
| E | 0 | 0 | 0 | 0 |

Table 1: Frequency distribution of the musical sequence

The frequency distribution is then used to calculate the chance of moving from one state to another by dividing the frequency of the new state, by the total states. For example when generating a new sequence and the current note is ‘D’, then the next note will have a 0.6 (3/5) chance to be another ‘D’, a 0.2 (1/5) to be ‘F#’ and the same chance for ‘E’. How the all of these different states interact with each other is shown below;

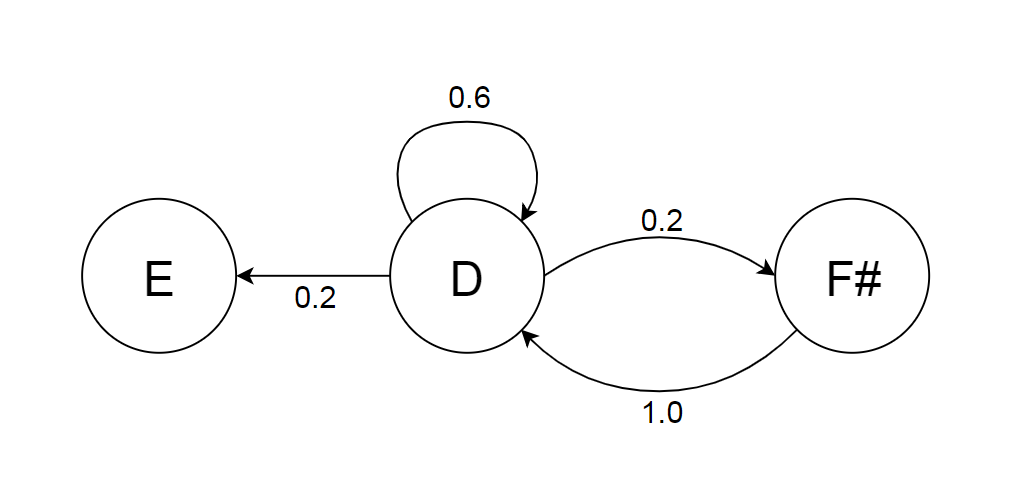


Figure 1: 1st order Markov chain for the musical sequence

This type of Markov chain is defined as a 1st order chain, as each new generated state is based solely on one step beforehand. A 2nd order chain would then be based on two steps before hand, and so on.

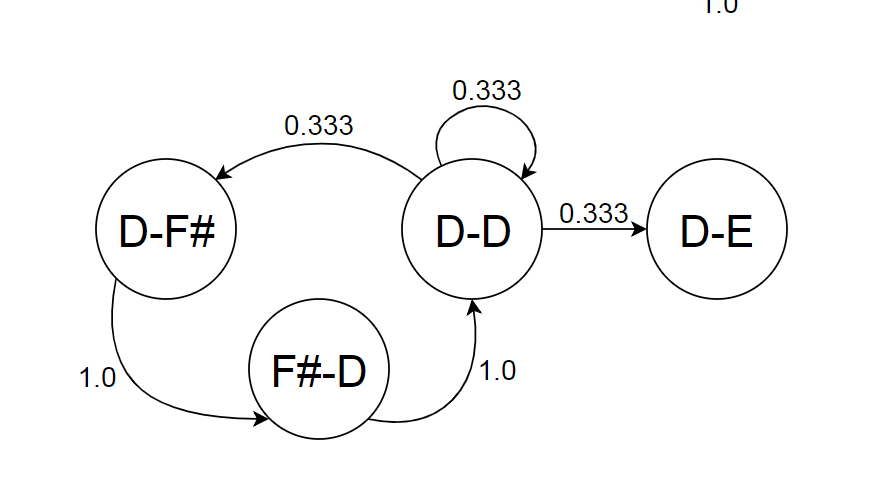


Figure 2: 2nd order Markov chain for the musical sequence

As the order increases the generated sequences have an increased level of similarity to the training data making the generated music sound less ‘random’, although this can also mean that the only possible sequence that can be generated is the trained data, which is not ideal when the purpose of the application is to generate new music.

Conversely to be too similar to the training data, a higher order chain can also produce states that were not even present in the training data, if this happens the application would require a method of overcoming this problem, such as stepping back through the data and then generating the failed sequence again (Snodgrass and Ontañón, 2013). Higher order chains may not even be worth the extra computing power that they require. Schulze and van der Merwe (2011) found that when presented with a human composed piece, music generated using a low order Markov, and one generated with a high order, the high order music was preferred the least. In regards to the randomness of the 1st order chains, Collins et al. (2011) found that generated music, based on Chopin’s Mazurkas, was difficult to discern from the real thing.

A recent example of Markov generated music was present in the game ‘Rise of the Tomb Raider’ (Crystal Dynamics, 2015) utilising the ‘dynamic percussion system’ (Intelligent Music System), this was originally developed by Brown (2012) for his PhD and utilises a combination of 1st order Markov chains, Genetic algorithms, and music theory to produce music at run-time. Following the other research on the topic, Brown found that the genetic algorithms took too long to generate music at run time, so this was done as the application loaded. While the original application was created to produce scored music for piano, it was utilised in ‘Rise of the Tomb Raider’ exclusively for the dynamic percussion, which would react to the various states that the player could find themselves in, from being completely hidden and the enemy not being aware of the player, to being engaged in combat. This dynamic drums were then set against pre-composed music (Lamperski and Tahouri, 2016).

Drums are a very common subject of procedurally generated music as they have a much lower level of variability in the number of note types that can be played (a standard drum kit has about 10 sounds, while a standard piano has around 80), this immediately reduces the complexity of any of the procedural techniques, coupled with the fact that drums notes have no musical restrictions on which notes can be played together, unlike melodic instruments. The length of a drum note also has less importance to the musical structure, as a drummer has much less control over a note’s attack and decay than a pianist. This allows rhythmic tracks to be much more ‘random’ sounding as they require a less cohesive structure than melodic sections.

## 2.2 - Creating Emotion

This sub-chapter will discuss the various musical rules which can be utilised in narrative elements of video games, and the ways in which music can be used to directly affect the player’s experience of a game.

When composing a piece of music with a particular emotional theme, the musical framework that it is built around is important for the narrative integrity of the piece. A clearer metric of the emotional state of a song is the valence and intensity (Schmidt and Trainor, 2001), these two terms are explained in the next subsections. These two attributes can be combined in varying proportions to create a wide range of emotions, for example high levels of both valence and intensity creates an exciting song, while high valence and low intensity creates a relaxing song (Cohrdes, et al., 2017). Mapping valence against intensity on a circumplex graph, first created by Russel (1978), shows the full range of emotions that can be expressed using this model, shown in Figure X.



Figure 3: Graphical representation of the circumplex model of affect, showing how valence on the x-axis (pleasant - unpleasant) and intensity on the y-axis (activation - deactivation), and how they combine to produce the various emotional states (Posner, Russel, ad Peterson, 2005).

### 2.2.1 - Valence

Valence is generally how ‘happy’ a song is perceived to be: high valence equating to high happiness. It is closely correlated with the musical concept of Brightness. This is metaphorically comparing music to luminosity: higher pitched sounds are associated with higher luminance. This was shown by Ludwig, Adachi, and Matsuzawa (2011), they required participants (both human and chimpanzee) to pick between a white or a black square while they played a note, both groups constantly associated higher pitches with brighter squares. However, this does not just apply to singular musical notes and can be applied to complete musical ideas. Bhattacharya and Lindsen (2016), showed that listening to music with a high valence made participants perceive a grey square as brighter than those who listened to music with a low valence, and Barbiere, Vidal, and Zellner (2007) found that participants associated happy songs with bright colours, and sad songs with grey.

One method of determining the brightness of a musical phrase is by calculating the spectral centroid (Schubert and Wolfe, 2005), this is the mean of the various frequencies that make up a musical idea. The various frequencies could be the resonant frequencies of a single note, when applied to the instrumentation of a piece (Eerola, Ferrer, and Alluri, 2012, and Jensen ), or all the notes in a whole musical piece. This explains why songs in a major key are commonly thought of as ‘happier’ than songs in a minor key as the spectral centroid of the minor scale is lower than the major scale’s key. This can be expanded as the major and minor scales are a subset of seven modes, these are; Lydian, Ionian (major), Mixolydian, Dorian, Aeolian (minor), Phrygian, and Locrian. These modes are made up of the looping pattern: three whole tones, a semitone, two whole tones, and a semitone, (it should be noted that a whole tone is equal to two semitones) although each mode has a different starting point, each mode also has a differing level of brightness, as shown in Fig. X

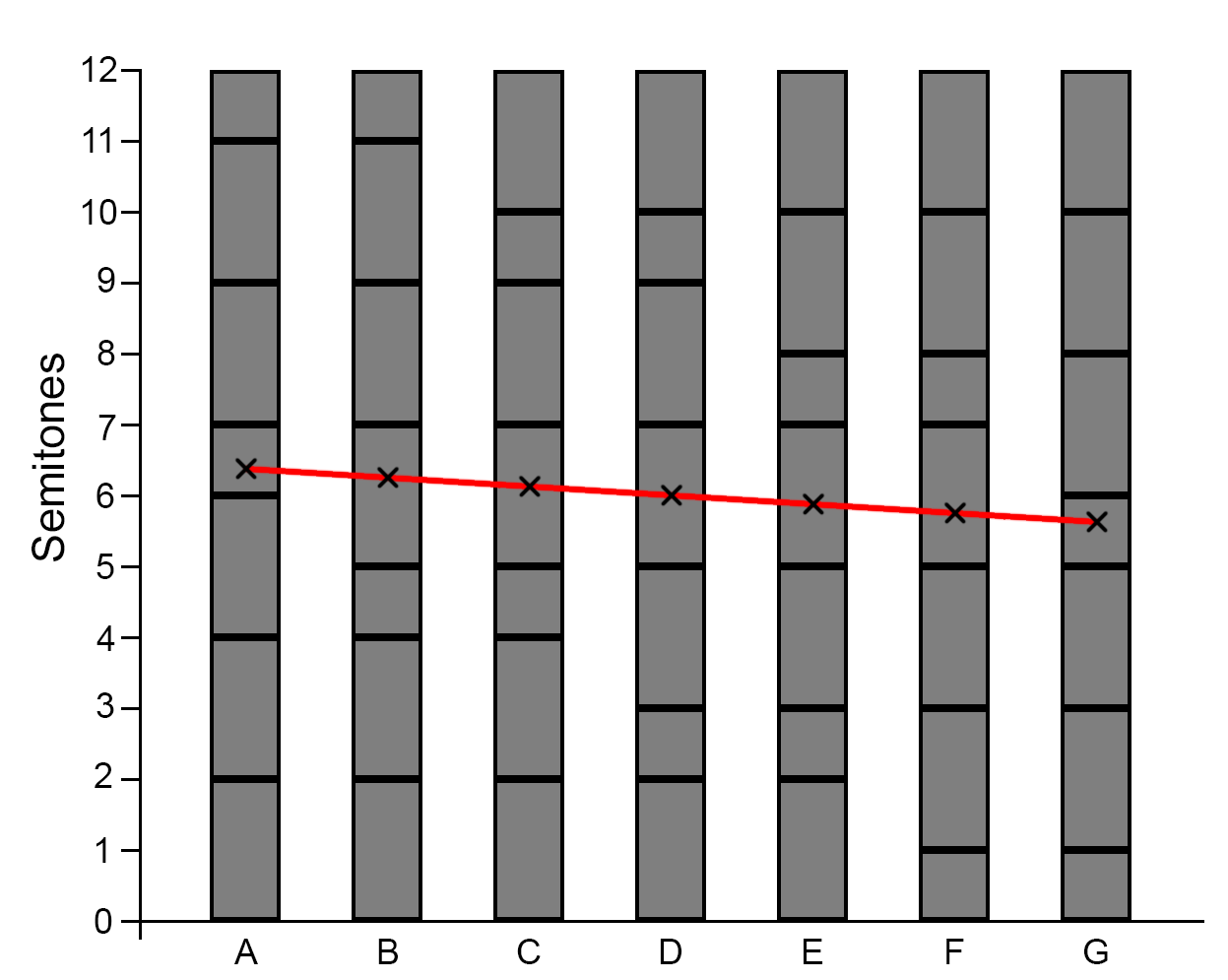


Figure 4: The semitone variation in the seven heptatonic modes, with their spectral centroid marked. Note the decreasing spectral centroid. A-Lydian, B-Ionian, C-Mixolydian, D-Dorian, E-Aeolian, F-Phrygian, G-Locrian

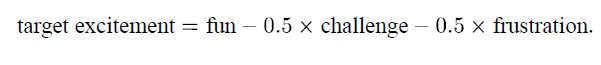
This tendency for higher pitched notes to be brighter than lower is consistent with the direction of a melody, that being a an ascending melody is perceived as brighter than a descending one (Collier and Hubbard, 2004)

### 2.2.2 - Intensity

The intensity of a piece of music mainly describes the various temporal variations that can be present (Droit-Volet, et al., 2013), such as the tempo or speed, the time of attack (the time it takes a note to reach its peak amplitude), the time of decay (the time it takes for a note to fade out), the rhythmic density (the amount of notes in a musical phrase), novelty (how much a the piece changes), and velocity (how loud a note is). The intensity of the music has been shown to directly correlate with the perceived arousal of the listener, or how alert they are (Dean, Bailes, and Schubert, 2011), although the energy (time of attack/decay, rhythmic density and velocity) has a greater influence than the tempo (North, Krause, and Sheriden, 2018). This was also shown in a study by Bramley, Dibben, and Rowe (2016), as they found that that tempo had little effect on the speed at which participants gambled. However, Mikutta, et al. (2013) found that tempo variations over the course of the piece would indeed increase arousal.

### 2.2.3 - Game Play as the Composer

Regardless of the method chosen to generate music, there needs to be some way that a game actively control the valence and/or the intensity of the music, an example of this is experience driven procedural content generation (EDPCG). Plans and Morelli (2012) used this for an infinite Mario Bros level generator, where they associated various actions in the game with frustration (time standing still and dying), challenge (alive time, time ducked), and fun (running time, coins collected, and monsters killed). These three variables were then combined using the following formula:



This excitement value was then used to control aspects of the music generation, such as the tempo, sparseness, and novelty (i.e. the intensity) and the musical scale used (i.e. the valence). While the authors found that the use of this system increased players’ enjoyment, they admit that they did not get enough testers to come to significant conclusions.

A similar experiment by Chan et al (2017) looked into how EDPCG can be used to inform the player of how well they are doing in a game. The game created to test this consisted of the player trying to find specific objects while avoiding hidden enemies. For this Chan et al used significantly less in game events to control the music, however this was to make the cause of the changes in the music more apparent to the player. They used; Tension (enemy proximity, low number of items collected, and low player health) and Progress (collectable proximity and high number of items collected). While this experiment also suffered from a low number of testers, the results found suggest that this method of procedural music control does indeed inform the player of how they are playing a game.

Another way that gameplay can affect the music generation is shown in an experiment by Mauceri and Majercik (2017). In this they created a music system based on a swarm algorithm reacting with a live performer playing a traditional musical instrument. The swarm would react to aspects of the performance such as; the amplitude would regulate the number of swarm members, the pitch would affect the general location of the swarm, and the musical cohesion would affect the swarm’s cohesion. The swarm would in turn interact with a granular synthesiser to produce the procedural music. For example, the difference in swarm member locations would determine the length of notes produced, and the mean Z location would affect the amplitude. While this method may not be appropriate in most game scenarios it would be interesting to have a similar system for a game where the antagonist was some sort of swarm or for it to be affecting the in-game wildlife.

## 2.3 - Musical Structure

### 2.3.1 - Theme

Another way to elicit an emotional response from music is to use themes (McCreary, 2018), these are ‘a prominent or frequently recurring melody’ (Oxford Dictionary, n.d.). These are used to musically represent characters and/or ideas, for example in ‘*God of War*’ (SIE Santa Monica Studio, 2018) the main character, Kratos, gets his own musical theme;

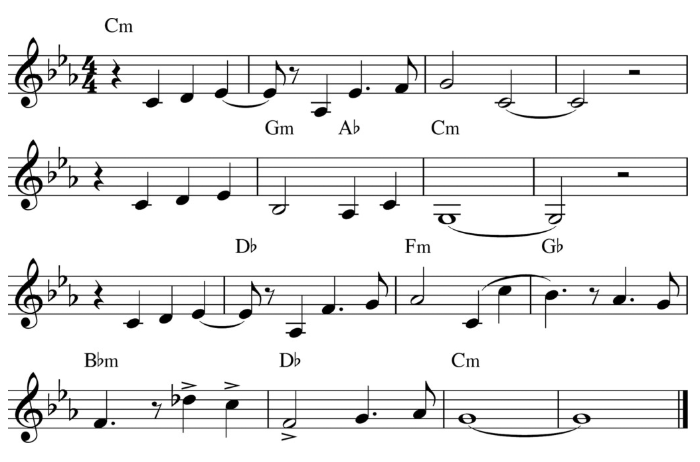


Figure 5: The Kratos Theme from God of War (McCreary, 2018)

In the game this theme is played, in some form, whenever Kratos appears on screen or his deeds are mentioned by other characters. As other characters in the game also get their own themes it allows for the music of a game to reflect what is happening in game, such as two themes being combined if two characters are interacting. A more famous example is featured in Star Wars Episode V ‘*The Empire Strikes Back*’(Kershner, 1980) where  ‘*The Imperial March*’ is introduced as the theme for the film’s antagonist, Darth Vader.



Figure 6: The beginning of The Imperial March, Darth Vader's Theme (Williams, 1980)

This theme mainly uses the Phrygian scale (low valence) to convey that Vader is the villain, coupled with the song’s short, loud notes (high intensity) it can be seen from Figure 4 that the piece conveys a stressful/tense feeling. Themes can also be used to show a character’s story arc. To continue the previous example of ‘*The Imperial March*’, in Star Wars Episode I ‘*The Phantom Menace*’ (Lucas, 1999), Vader is shown as a young innocent child, Anakin Skywalker. Here the theme uses the same chords as ‘*The Imperial March*’, but in their major variation, showing that while it is the same character he is not the villain yet.

### 2.3.2 - Leitmotif

A theme can be broken down into a short phrase know as a leitmotif, this can be used when playing a character’s entire theme may not be appropriate, but they work in the same way a theme does in terms of narrative meaning. For example Kratos’ leitmotif is the first three notes of his theme;



Figure 7: Kratos' theme, his leitmotif is repeated three times, highlighted in red

As is shown in Figure X the leitmotif is repeated three times in his theme to make it obvious to the listener that this is the case, in the full score for this piece the bass line is also playing the leitmotif as well. For ‘*The Imperial March*’ the leitmotif is also the first line;

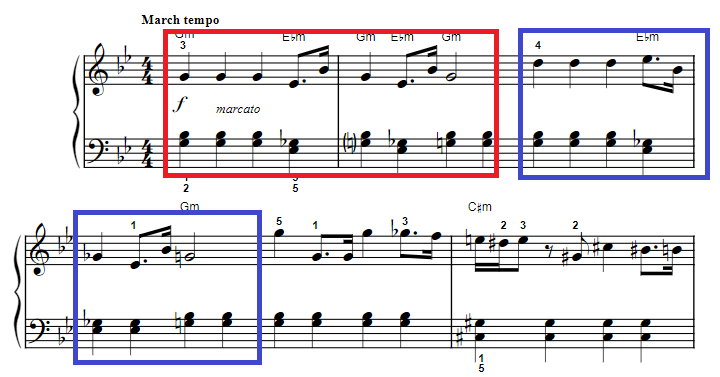


Figure 8: The first line of 'The Imperial March' with the leitmotif highlighted in red, it is then repeated (shown in blue), with some variation in pitch, however the rhythm is consistent

A procedural application would then have to have some sort of procedure for generating a theme, this could be taken directly from the output of the procedural technique chosen, however as shown above leitmotifs are often repeated throughout a theme. When composing a theme there are two main structures that are used, the Period and the Sentence.

### 2.3.2 - Period

<http://openmusictheory.com/period.html>

The Period is split up into four sections, the first two are collectively named the Antecedent, while the second two are the Consequent. Each of these are then made up of two parts, a basic idea and a contrasting idea. For both the Antecedent and the Consequent the basic idea is the same, although some tonal variation can happen, and the contrasting ideas are both different. An example of this is Kratos’ theme as can be seen from Figure X below.

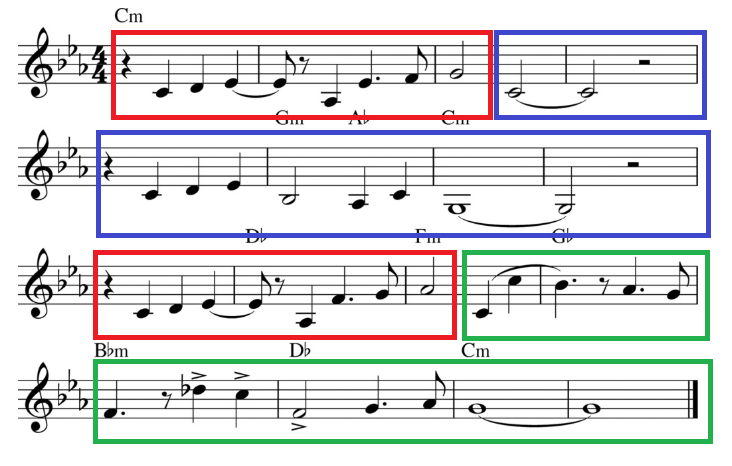


Figure 9: The period structure of Kratos' theme, the basic idea outlined by red, with the two contrasting ideas in blue and green

Bit about how the chord patterns are structured, end of antecedent is weak cadence, end of consequent is strong. Although im not doing this, cause time contraints.

### 2.3.3 - Sentence

<http://openmusictheory.com/sentence.html>

The Sentence is of similar idea, however in the first half, named the Presentation phase, the basic idea is repeated twice, and again the repetition need not be homogenous with the first. The second half, named the Continuation, expands upon the ideas given in the basic idea, and leads the piece to its conclusion. The Imperial March is an example of this type of structure, shown below in Figure X

Add bit about fragmentation/liquidation/those things?



Figure 10: The Imperial March with its sentence structure highlighted. The first basic idea in red, the second basic idea in blue (note the tonal variation), and the continuation in green

## 2.4 - Summary

The chosen procedural technique for this project was 1st Markov chains as they are well suited to pattern replication. This was used for melodic and rhythmic sections. The main melodic sections can be generated either in Period or Sentence form. EDPCG was used to control the intensity and valence of the produced music. For the melodic sections a set of rules were created to make sure the composed melodies were in the correct scales (decided upon by the valence). A simple game was produced to allow enough scenarios to showcase that intensity/valence variation.

NEED TO SAY THIS WILL BE EXPLAINED IN METHODS?

ALSO REWRITE THIS TO MAKE IT SOUND BETTER