

Generating Musical Sheets for First-Sight Reading Exercise

Frederico Freitas
Instituto Superior Técnico,
Universidade de Lisboa and
INESC-ID, Lisboa
Edifício IST-Taguspark, Av. Prof.
Aníbal Cavaco Silva, 2744-016
Porto Salvo
frederico.freitas@
tecnico.ulisboa.pt

Rui Prada
Instituto Superior Técnico,
Universidade de Lisboa and
INESC-ID, Lisboa
Edifício IST-Taguspark, Av. Prof.
Aníbal Cavaco Silva, 2744-016
Porto Salvo
rui.prada@
tecnico.ulisboa.pt

Pedro A Santos
Instituto Superior Técnico,
Universidade de Lisboa and
INESC-ID, Lisboa Edifício IST-
Taguspark, Av. Prof. Aníbal
Cavaco Silva, 2744-016 Porto
Salvo
pedro.santos@
tecnico.ulisboa.pt

Luís Coelho
Lusomusic Lda
Rua da Chieira nº 512,
2435-152 Espite
luis.coelho@lusomusic.pt

Abstract

This article describes how we tackled the problem of generating appropriate musical exercises, destined for musical students to develop and practice sight reading skills, under the context of the JAMP project, a serious game aimed to complement musical learning, a project financed by community funds from the EU through the Programa Mais Centro (QREN). We've submitted our implementation for testing with experts with background on music teaching, who found the generated content to be musically correct and pleasing sounding.

Keywords

Procedural Content Generation, Music Generation, Partiture, Sight-reading, Game AI.

1. INTRODUCTION

The JAMP project is a serious game, aimed to serve as a musical learning complementary tool, for aspiring musicians who want to learn the piano, or similar keyboard-based instruments. Developed for iOS and the iPad, it consists of a series of games aimed to develop different competences, such as rhythm reading, hand dexterity and musical partiture interpretation. One of the core competencies targeted by this game was for students to be able to reinforce their skill at first sight reading, meaning, grabbing a fresh musical sheet, and interpret its contents without any form of previous rehearsal. This means, that unlike other exercises, the content has to be new every time the student goes through it. Also, it must serve meaningful pedagogic purpose.

Looking at related work in the field of music composed by computers, both related to generating musical sheets or actual sound synthesis, we can usually discern two branches for the creation process, supervised (music composed with the help of a computer), and autonomous (fully composed by the machine). Inside these branches, several methods have been considered. Methods such mathematical models, knowledge-based systems, gram-

mars, machine learning, systems based on genetic algorithms or even hybridizations of the above. [1][2][3][4][5][6][7].

However, the purpose of this work follows a direction apart from most works related to the topic, since we not only aim to provide an adequate sounding score, but it must be a tool aimed for learning and training sight reading, which meant that our compositions probably followed a supplementary set of requirements.

An analysis of the parameters that compose such content, followed by how we create a meaningful learning progression and finally how given these parameters we're able to generate musical exercises that serve the game's intended purpose are the subjects tackled in the third and fourth sections of this document.

2. RELATED WORK

We have researched a few musical learning platforms to evaluate how our development context (videogame with adaptive AI aimed towards the pedagogic aspect) fares against the current solutions. Looking through it, we can split them into two segments, since there was not a product that offered what our solution aims to, an artificial

intelligence architecture that generates fresh content following the player's current level of proficiency, integrated into a game experience.

2.1 Music Learning Games

Inside this segment, given the intended learning experience that these games aim to teach, we have games that are geared towards musical theory learning (they seek to teach musical theory, such as clefs, modes, rhythm figures), and games which deal with the more practical aspect of music, such as learning how to play a particular instrument.

A good example of the first segment would be Tonic Tutor (<http://www.tonictutor.com/>), a collection of simple games that test different aspects of musical theory such as note reading, ear training and musical theory. The most interesting asset provided by the platform, is the possibility for teacher to register students, assigned lessons by tailoring the minigames to test particular domains of knowledge and keep track of their progress. However, the generation of all the content has to be supervised by a human, since the platform prefers to leave in the hands of the teacher what is crucial for the student to learn, given his perceived level.

On the opposite end of the spectrum, Rocksmith (Ubisoft, 2011), a game for PC, PS3, PS4, Xbox 360 and Xbox One, allows the player to learn how to play guitar and bass, alienating most theoretical knowledge from him, by resorting to a graphical representation that very closely matches the instrument in question. It also features a progression that system that allows the user to learn and practice the instrument by going through a large song list that is ordered by difficulty. A player learns a song in Rocksmith by dynamically adjusting the difficulty to his skill level, and then slowly introduces more notes and phrases until the song is being played note by note.

However, it does not feature any form of generating content for First-Sight Reading purposes. All the material is hard loaded into the game and does not intend to test the target competence that aims our work.

2.2 Other Musical Software

We have also explored a few examples outside of the videogame realm searching for software that share some of our objectives for this work.

We start with EarMaster (Miditec, 1996). It provides a plethora of exercises divided by courses for ear training, rhythm training and sight reading. It features a more ample selection of input methods; users can perform the exercises by voice, clapping or using a midi controller. Users also have access to several statistics that detail their performance and progression in their intended areas and a degree of customization regarding the exercises themselves. However, it does not provide the ability to generate an exercise from scratch, relying instead on selecting specific examples from its database, given supervised input from the user.

A similar approach, but deviating towards music theory learning is the one given by Auralia, part of the Sibelius software suite (Avid Technology, 2003). Users are able to select courses that explore five major topics: Intervals and Scales, Chords, Rhythm, Harmony and Form and finally Pitch and Melody. Each one of these topics provides several sub-topics that can be explored individually by the user and each exercise can be customized according to the player's necessities, for example, if he wants to practice distinguishing a 4th interval from a 5th one, he can tailor an exercise to only include such intervals.

While these two examples provide ample forms of customization, they leave in the player's hands (or the teacher who's supervising) the decision on how they should carry their progress. Our implementation aims to make that process completely transparent to the user, by defining core parameters of learning proper First-Sight reading, and organizing them into a progression that relies solely on how he progresses through the suggested levels.

They also provide a limited level of procedural content generation. Most rely on already pre-established exercises that are filtered and selected according to the player's needs. Our approach tries to provide a different score every time the player attempts an exercise, even if the generation parameters are essentially the same. There exists a plugin for the Sibelius software named Melody Generator, made by the Music Transcriber. This plugin allows the generation of melodies from scratch, using a set of parameters such as featured pitches and rhythm figures, mode and tonality. While functionality wise, our approach resembles this solution, the generation process is devoid of any pedagogic supervision. The user generates the melody, and while he can use it to practice First-Sight reading, among other uses, it doesn't provide an underlying learning progression to guide both the player, and the generation process.

3. PROBLEM STRUCTURE AND PARAMETERIZATION

The exercises have to comply with two major objectives. First and foremost, they have to serve its pedagogic purpose, meaning that it must satisfy a set of restrictions, based on parameters established by the music teachers we've worked with, and also offer learning progression, which means that these exercises become increasingly more difficult as the student clears the content. Later in this section, we will explain these parameters to deeper detail. Second, they need to sound 'right', as in, the notes being played need to be pleasing to hear, up a to certain degree.

A musical sheet has two major components. The rhythm of the song/track, which explains to the interpreter the tempo and timing for he will play the notes detailed in the melody, essentially describing the keys of each note, hence, where are they played on the keyboard. Usually, for keyboard based instruments, a musical sheet separates the hands, having a distinct voice for each. Exercises that involve more than two hands are out of the scope of the application.

Informal interviews with the musical experts were conducted in order to understand the basic elements that make music, and try to convert these, into parameters that can be used to generate a musical sheet, and scaled into a learning progression system.

3.1 Rhythm Analysis

Three parameters make part of the rhythm aspect of a musical score. Time signature, tempo and the figures used.

3.1.1 Time Signature

The time signature details how many beats are in each measure of music and what figure represent the beat. An example would be a common 4/4 time signature, which means that each measure has a total time of four (represented by the top number) quarter note beats (represented by the bottom number) worth of time. Although there are a vast amount of different time signatures, a few have been selected and ordered by difficulty, task handled by the musical teachers that support the project.

3.1.2 Tempo

Tempo is how fast a beat is played. If a music score has a tempo of 60, that means that each beat is worth a second in real time. If it is 120, then the interpreter plays at a rate of two beats per second. Difficulty of execution is directly proportional to the tempo.

3.1.3 Figures

Rhythm figures are used to represent when and how a note is played (or the absence of playing). Several of these are paired with other figures, creating patterns which are going to be important on generation process. How these figures are paired and ranked in terms of difficulty is related to the time signature used by the exercise.



Figure 1: A - example of a 4/4 time signature. B- Indication of Tempo, indicating that the quarter note is played with a speed of 113 beats per second. C- An instance of a Figure, namely an eight-note, worth half of a quarter-note.

level	ME	MD	BPM	TS
1	1 4	1 4	60	4 4
2	1 2 4	1 2 4	60	4 4
3	1 2 4 5	1 2 4 5	60	4 4
4	1 2 4 5 44 45	1 2 4 5 44 45	60	4 4
5	1 2 4 5 44 45	1 2 4 5 44 45	60	3 4
6	1 2 3 4 5 6	1 2 3 4 5 6	60	4 4
7	1 2 3 4 5 6 8	1 2 3 4 5 6 8	60	4 4

Figure 2: Excerpt from the table that aggregates Tempo (BPM), Time Signatures (TS), and the figures to be used in each hand (ME and MD) into levels of difficulty, to be used later during the generation procedure.

3.2 Melodic Analysis

3.2.1 Key

Most people are familiarized with the steps from a common “C” scale. This is one of the instances of the musical system referred as key, or tonality, which establishes relationships between pitches or keys for the melody. The keys found inside a specific tonality, save a few very rare exceptions which are not covered by the scope of the desired exercises, are the keys that will figure in the generated musical score.

3.2.2 Melodic Quality

Every level will feature a musical loop that serves as backing track for the exercise. This not only enriches the experience of the user as emulates playing with a band, but also sets a guideline for our generated musical score to become more audibly pleasing. It is entirely possible to create a musical piece that abides by all the rules, yet it can sound disconnected and void of purpose.

Each loop, details what keys should be placed at given times in our composition to make better harmonies. This information has been again compiled for each loop by its actual composer in table, to be used during our generation procedure.

3.2.3 Hand Technique

The distance between notes is called an interval. We call a movement the way we use our fingers to traverse a certain interval on the keyboard. There is not a direct mapping between an interval and the fingers used to play it, in fact, different finger combinations can be used to play the same sequence of keys. Context of the movement dictates which are better suited for the occasion, namely the scope of the notes reachable during the melody sequence.

Hand Technique determines the maximum difficulty of the movements applied during the exercise. Further description of this concept is made in the next section, once the generation algorithm is explained.

3.2.4 Hand Synergy

The way that the notes from each hand relate to each other is described as the synergy between them. Musical experts have categorized this synergy into seven different categories. From easier to hardest, they are described as:

- One hand: means that only one of the hands executes the exercise.
- Alternating: Both hands are alternating between playing and resting, meaning that when a note is playing in on hand, the other rests. A two measure count is given before the hands can alternate again.
- Mirror: Both hands play the exact rhythm, and the notes are given in a way that the finger movements mirror from one hand to the other.



Figure 3: Mirror hand synergy.

- Oblique: One hand plays a line, while the other sticks to one key, although it can play a different rhythm.



Figure 4: Oblique synergy between hands.

- Parallel by an octave: Both hands play the same rhythm, and the hands follow the same movement direction on the keyboard. The interval between a note from the left hand and a right one is always an octave, which makes both hands play the same key, but in a different octave. Generally, finger notation is done in reverse from one hand to the other (right hand plays a key with the first finger while the left plays it with the fifth, right with the second while the left with the fourth, and so on).
- Parallel: Same as parallel by an octave, but the interval between notes in each hand doesn't have to be an octave.



Figure 5: Parallel motion example. There is always a 6th interval between hands.

- Contrary: In this mode, any kind of movement is allowed.

3.2.5 Keyboard Scope

Every exercise targets a specific area of the keyboard for each hand which we call scope. The more keyboard area is covered; it is assumed that the exercise will be generally harder.

Level	Left Hand		Right Hand	
	Low	Up	Low	Up
1	A3	C4	C4	E4
2	F3	C4	C4	G4
3	E3	C4	C4	A4

4	A2	E3	A4	E5
5	A2	C4	C4	E5
6	E3	C4	C4	A5
7	E3	E4	A3	A5
8	C2	E4	A3	C6
9	C2	G4	F3	C6
10	A1	G4	F3	E6
11	A1	G4	F3	G6
12	A0	G4	F3	C7
13	A0	G4	F3	C8

Figure 6: Table detailing several scopes, with an assigned difficulty for each. The values are a letter representing the step, paired with an octave, which maps into a key from the piano.

The analysis was done in two steps. First, like mentioned before, we sat down with the experts and divided the concept of musical score into the characteristics that we refer as parameters and analysed each until we came up with attributes and values to describe each. Then, they ordered each into levels, attempting to express them as individual axis for learning progression (Fig. 2, 6 and 7), and eventually aggregating all into a meta-table (Fig. 8), where each line corresponds to an actual list of values for each parameter, to be used as a generation basis for a given exercise/level.

Level	HT	SN L	SN R	HS	MI
13	1	1	1	3	6
14	1	1	1	4	6
15	1	1	1	5	6
16	1	1	1	6	6
17	2	1	1	3	7
18	2	1	1	4	7
19	2	1	1	5	7
20	2	1	1	6	7
21	2	1	1	1	8
22	2	1	1	2	8

Figure 7: Excerpt of the table that collapses Hand Technique (HT) and Hand Synergy (HS) parameters, along with the maximum amount of simultaneous notes that can be played in each hand (SN L and SN R) and the maximum interval (MI) allowed in a single movement into a single meta-parameter that we refer as Technique.

Level	SD	RD	TD	TOD
1	1	1	1	1
2	1	2	2	1
3	2	2	3	1
4	2	3	3	1
5	2	4	3	1
6	2	5	3	1
7	2	6	3	1
8	3	6	3	1
9	3	7	3	1
10	4	7	3	2

Figure 8: Excerpt from the meta-table that aggregates the values for the learning progression of all parameters, and maps them into levels. Keyboard Scope (SD) (Fig. 6), Rhythm Difficulty (RD) (Fig. 2), Technique (TD) (Fig. 7) and Key (TOD).

4. IMPLEMENTATION

Our hypothesis for the musical score generation starts by separating the rhythm component from the melodic component. This decision makes sense since rules melodic construction do not conflict with those for rhythm, from both a musical standpoint, but also from a pedagogic perspective.

4.1 Rhythm Generation

We started by using the single rhythm figure as the basic element for our exercise rhythm section. However the number of combinations possible between figures is enormous and not all of them are appropriate. Some of these combinations are not optimal regarding some time signatures; others while correct according to formal rules, have alternatives which are better suited for reading. Also, these combinations have different degrees of difficulty which are not easily extrapolated from each of its elements alone. So we asked the musical teachers to arrange a list of these figure combinations which we call rhythm cells. A cell is represented by one or more rhythm figures, and unless one of its elements lasts longer than a quarter-note, then save a few exceptions, the combination doesn't last longer than a quarter-note.

We've also asked the musical experts to provide a table where these cells would be grouped and ordered according to a difficulty level, and related to a desired time signature. Generation is fairly straightforward. For the intended difficulty, we pick the associated list of cells, and proceed to randomly pick a cell to fill each of the exercise's composing measures, as long as two restrictions are not violated. First, we make sure that if a particular cell is selected, it won't exceed the remaining time a measure has. Secondly, we avoid placing cells that contain pauses during strong beats, sections of the measure which have great importance on how consistent or "well" the music sounds. At a base level, a group of selected cells have the same probability of being selected, but by suggestion of the musical experts, we've lowered the chance of placing pauses on an exercise, so they remain interesting for the player.

A rhythm line is generated for each hand, although, given an intended hand synergy parameter, the generated line for the left hand might be changed.

4.2 Melodic Generation

The basic element of a generated melodic line is the movement. Using a single note or pitch presents the same hurdles found when using rhythm figures as the basic generation unit for rhythm parts. It is hard to establish relationships between notes using an unsupervised method and assign a proper difficulty rank. Also, the same note sequence can be done using different finger combinations.

The experts created a table that contains all the possible hand movements done in a keyboard based instrument. A movement has the following attributes:

Finger A	Represents the first finger involved in the movement.
Finger B	Details which finger ends the movement.
Interval A	Refers to the orientation of the movement (upwards, downwards).
Direction A	Refers to the orientation of the movement (upwards, downwards).
Interval B	Represents a set of interval candidates for the next movement, which are compatible with the current one.
Direction B	The expected orientation of the next candidate movement.
Difficulty	How technically challenging the movement is, in a scale from 1 to 4.
Inversion	If the next movement is supposed to have a different orientation from the current one, then this flag is signalled as true;
Minimum Scope	Defines the minimum sum of the intervals that a chain of movements must target, before switching the direction.
Maximum Scope	Defines the minimum sum of the intervals that a chain of movements must target, before switching the direction.

Figure 9: Attributes related to hand movement.

The objective now is to chain these movements into creating a melodic line for each hand.

To do so, we use a depth-search approach. The generation process goes like this:

- 1- We filter out the movements that are above the intended maximum interval and movement technique difficulty.
- 2- We pick a root node. In an optimal case, we pick one from the nodes in which the movement starts with the first finger.
- 3- Fetch all the possible children, applying all the hard restrictions.
- 4- If the children are empty, return to the previous level, and pick another node to explore.
- 5- Otherwise, order them, using a heuristic function. Said heuristic simply prioritizes nodes that represent movements which will lead to notes that will sound good, according to the chosen background loop before generation. If there are several nodes that fit these criteria, randomly pick one for the next step. Same applies if there are none (but still comply with the hard restrictions).
- 6- Explore the selected node, repeating from step 3.

A solution will be found if we manage to reach the desired depth of the generated tree. The depth is given by the actual playable notes generated for the rhythm section of the exercise.

When generating the children of a node, the following restrictions are applied:

- The starting finger (Finger A) of the next candidate child must be equal to its parent destination finger (Finger B).
- The next candidate movement interval must not lead to a note that is out of the key boundaries intended for the exercise.
- If the child movement implies a direction change, then all the movements made before the previous direction change, must have an interval sum within the Minimum Scope and Maximum Scope attributes found in the movement that started aforementioned direction change. If this condition is not met, then that candidate node is not part of the children list.

This approach was selected because the generated trees are usually deep (best case scenario, given that the loops amount to 17 measures, and at least one note is needed for each measure, results in a tree with a depth of 17. Some cases, with rhythm cells made of 4 or plus figures, it can go over 60), and the branching factor is, at worst, 23 which represents the maximum number of possible movements given any starting finger. Given these values, a more informed search such as an A*, would represent greater costs in both speed, and especially memory, concerns that are more flagrant since the final game is target-

ed towards mobile platforms such as the iPad. It also helps the depth-case that all the reachable nodes that inhabit the target depth represent possible solutions. Potentially not selecting the best sounding melodic line was an acceptable compromise given the aforementioned constraints, validated in preliminary tests made with experts.

The last part is to join both the rhythm and the melodic movements generated for each hand. We map the keys for the movements according to the intended tonality, and change the rhythm or melody to suit the desired hand synergy type for the exercise. Simultaneous notes (chords) are also introduced having the current key serve as the root, depending if the exercise parameters deem so. Basically, there is a probability that simultaneous notes can be generated when mapping a particular note. The added notes respect the tonality. The probability for the event to happen was inferred and tested in cooperation with the musical experts. Ties between notes are out of the scope of our implementation, but might be feature added at a later phase.

5. RESULTS AND SAMPLES

The next two figures are samples of generated musical sheets in the game context. From the data collected preliminarily, the generation process is fairly lightweight, adding values around the one second mark to the general level loading, for trees with a depth around 60 (60 notes generated for each hand). Save for a few particular cases, the combination of parameters and their respective values result in valid generations.



Figure 10: An example of a generated score, for practicing oblique hand synergy.



Figure 11: Example of a generated exercise focused on alternated hand practice.

This was confirmed with preliminary tests made with the musical experts that accompany the development of the game have given credibility to the approach. So far, the generated exercises follow the rules of musical writing, and sound good when paired with the available backing track, minus a few occurrences. We owe this to the search approach used in the generation of the melodic lines.

6. CONCLUSIONS AND FUTURE WORK

Further work regarding analysing new metrics or even trying a more informed search algorithm will be done to improve the shortcomings described in the previous section. We also consider that very limited testing has been (until the date of submission of this article) with actual aspiring musicians, leaving the pedagogic aspect of the project still unproved.

7. ACKNOWLEDGEMENTS

The good folks from Wingzstudio.lda and Present Technologies, co-developers of the JAMP project. Manon Marques, Mariana Godinho and Fernando Loura for the countless hours of patience, teaching, testing and understanding. This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 and also by community funds from the EU trough the Programa Mais Centro (QREN).

8. REFERENCES

- [1] Fernandez, JD; Vico, F (2013). "AI Methods in Algorithmic Composition: A Comprehensive Survey." (PDF). *Journal of Artificial Intelligence Research* 48.
- [2] Dubnov, S., G. Assayag and R. El-Yaniv. (1998). "Universal Classification Applied to Musical Sequences." *Proceedings of the International Computer Music Conference*. International Computer Music Association.
- [3] Cope, D. (1987) An expert system from computer-assisted composition, in *Computer Music Journal*, 11(4).
- [4] Cope, D. (1991). *Computers and musical style*, Oxford University Press.
- [5] Assayag, G., C. Rueda, M. Laurson, C. Agon and O. Delerue. (1999b). "Computer Assisted Composition at Ircam : PatchWork & OpenMusic.", *Computer Music Journal*, 23:3.
- [6] Brooks Jr., F., Hopkins Jr., A., Neumann, P., and Wright, W. (1993). An experiment in musical composition, In Schwanauer and Levitt, editors, *Machine Models of Music*, pages 23– 40. MIT Press.
- [7] Lidov, D. and Gambura, J. (1973), A melody writing algorithm using a formal language model, in *Computer Studies in Humanities*, 4(3-4).