Scriabin in the Digital Realm

Tamara Duplantis
University of California Santa Cruz
Santa Cruz, United States

Kyle Gonzalez
University of California Santa Cruz
Santa Cruz, United States

Cassandra Ravenbrook
University of California Santa Cruz
Santa Cruz, United States

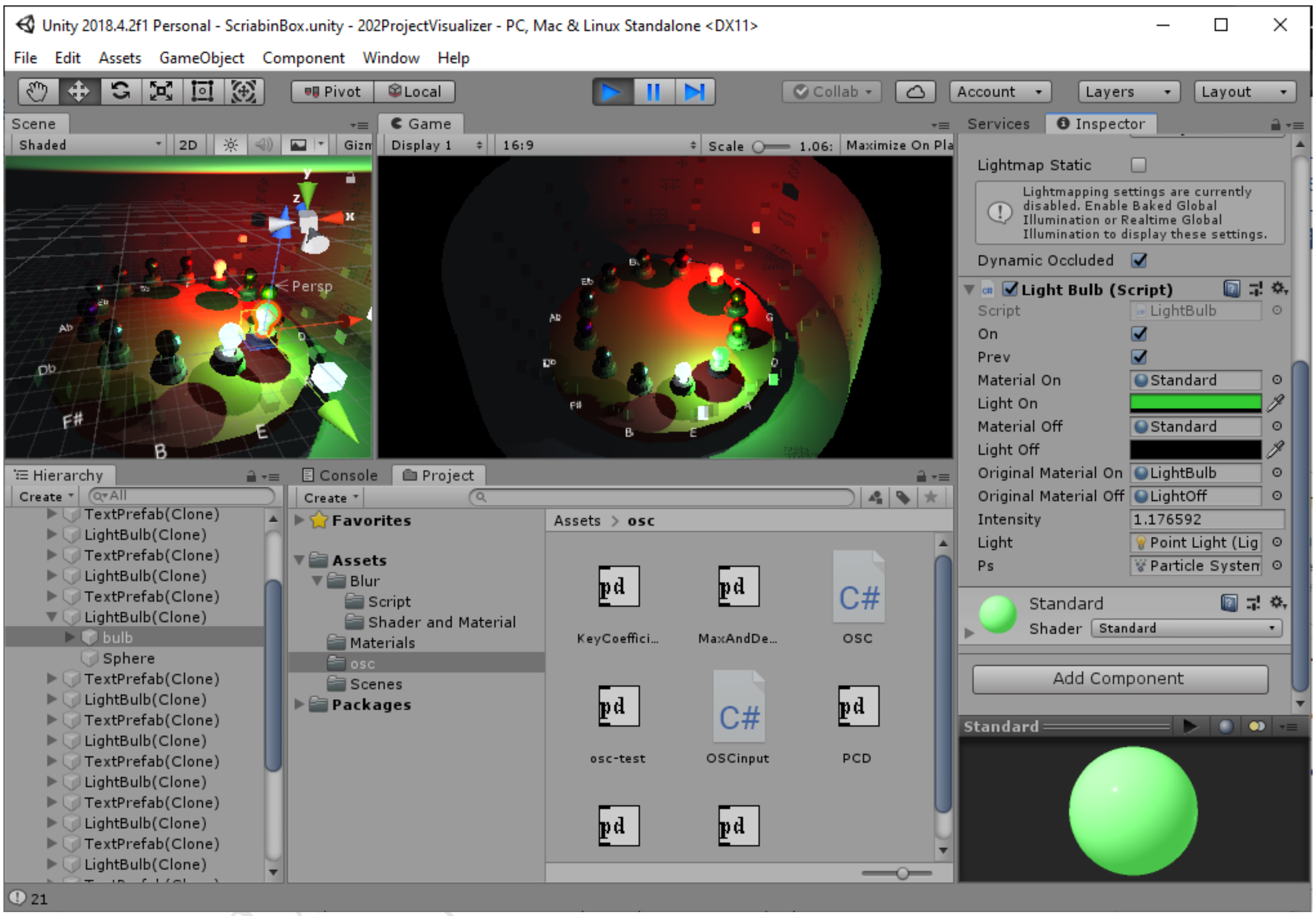

Figure 1: Unity Project View

ABSTRACT

In this paper we use the Krumhansl-Schmuckler model and algorithm, implemented in Pure Data to drive a music visualization application inspired by the key color system devised by Alexander Scriabin. MIDI input is run through our key tonality model and into our visualizer, created with the Unity game engine, which renders a three-dimensional space. The lighting and materials of objects within that space are directly affected by the output of our key tonality model and MIDI information.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Alexander Scriabin, key detection, Krumhansl-Schmuckler model, music-visualizer, Pure Data, Unity

ACM Reference Format:

Tamara Duplantis, Kyle Gonzalez, and Cassandra Ravenbrook. 2018. Scriabin in the Digital Realm. In *Computational Media Research.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Our visualizer is modeled after Alexander Scriabin's *clavier à lumières*, a modified keyboard instrument in which key presses trigger the illumination of colored light bulbs. The colors associated with each key were mapped in accordance with the composer's experience of sound-to-color synesthesia, a perceptual phenomenon in which sounds are perceived as color depending on their specific qualities. In Scriabin's case, his associations made correspondences between the color wheel and the circles of fifths. Each case of synesthesia is subjective to the individual's own perception, with very few associations being standard across all synesthetes, yet we chose to focus our project on Scriabin's synesthetic associations for two major reasons. The first is that these associations are a very well-documented case, with articles and interviews published within the composer's lifetime detailing the specifics of his synesthetic associations, as well as the existence of a physical device, the *clavier à lumières*, designed by Preston S. Millar and commissioned by the composer to display these colors. The second is that, due to the *clavier à lumières* being an integral component of the presentation for Scriabin's symphonic work *Prometheus: The Poem of Fire*, the development of a virtual facsimile of Millar's unique instrument is a worthwhile goal in preserving this early multimedia instrument, as well as allowing performers to recreate Scriabin's work as it was originally intended.

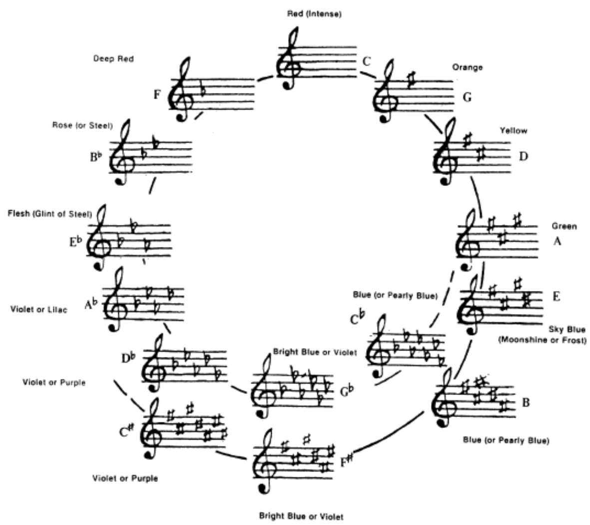

Figure 2: The circle of fifths labeled with their associated colors based Leonid Sabanayev documentation.[3]

2 PREVIOUS VISUALIZATION AND PERFORMANCES OF SCRIABIN

We could not find any digital visualizers that directly take inspiration from Scriabin's work, however we found contemporary works within the physical domain that apply his principles. Nick Kanozic, musician, composer, visual artist and chair of the music department at the Oakland School for the Arts, in "Light Organ - Scriabin No 1" Kanozik applies Scriabin's color system to his recreation of the light organ. [4] In "Scriabin's Prometheus: Poem of Fire" [6] Gawboy, Shimada, Townsend attempt to create Scriabin's vision using modern lighting technologies and the use of the Yale Symphony Orchestra.

Our work is more closely aligned with that of Kanozic. Our digital visualizer is a purely procedural artifact and uses Scriabin's established associations to interpret its input rather than being an instrument of sound to be arranged for and played.

3 KEY TONALITY MODEL

In order to interpret notes that were played in real time, we implemented a version of the Krumhansl-Schmuckler key-finding algorithm[5]. This algorithm was developed in the disciplinary context of cognitive science, and it is situated among other algorithms for key determination that come out of work in artificial intelligence. Indeed, one of the main motivations that drove the development of the algorithms was a desire for automated musical analysis in the realms of harmonic structure, Schenkerian analysis, and pitch-set analysis – all music theoretical constructs developed before the 60's but which had then been operationalized in cognitive science and artificial intelligence.

One of the main research orientations in this field at the intersection of music theory and computational models which directly organizes the assumptions of the Krumhansl-Schmuckler algorithm is an approach to the cognitive interpretation of music through a hierarchical model of pitch classes in relative stability or resolution, deeply influenced by the music theoretician Heinrich Schenker. The Krumhansl-Schmuckler algorithm is based on such a model of tonal hierarchies as developed by Krumhansl and Kessler a few years earlier. The specific values of the model were derived from experimental data, where researchers asked how well a certain set of notes was completed or resolved by different pitch class as a series of "probe tones." Suffice it to say, there are many assumptions underlying such an experiment and the resulting model. One justification provided in the study is the psychological principle of the relative importance of certain conceptual and perceptual objects – think of Gestalt closure applied to a series of tones. Thus the researchers were in effect testing the underlying music theory as a hypothesis supported by psychological principles.

The algorithm for determining key operates as follows. The duration of each pitch class is measured over some time span, either number of beats, or seconds, or simplified in our case, order of pitches. Correlating these analyzed pitch class durations with a set of 12 major and minor "profiles," one for each key, where each profile contains the experimentally-derived values corresponding to each pitch class that measure relative stability within a certain key. The resulting correlation coefficients thus measure how likely a set of pitches in some time span belong to any major or minor key. This relative rather than absolute determination of key reflects the phenomenon of a listener entertaining multiple key "hypotheses."

Because we could find no implementation of the Krumhansl-Schmuckler algorithm, we programmed our own version in a specific build of Pure Data called Purr Data in order to use specific libraries for pitch analysis of MIDI data. We take in MIDI data and feed it into a custom abstraction *PitchClassKeyCoefficient* which returns a vector of correlation coefficients, one for each pitch class, where the coefficients for each the major and minor modalities of a pitch class are combined, reflecting a larger-scale model ontology where modal changes do not affect key. After going through an external pitch-determination object, we feed the resulting pitch class value into a series of custom objects that produce artificial "durations" of each pitch class based on the order they were input. These durations then go through a dense matrix of mathematical manipulations in order to return the correlation coefficients.

4 USER INTERFACE

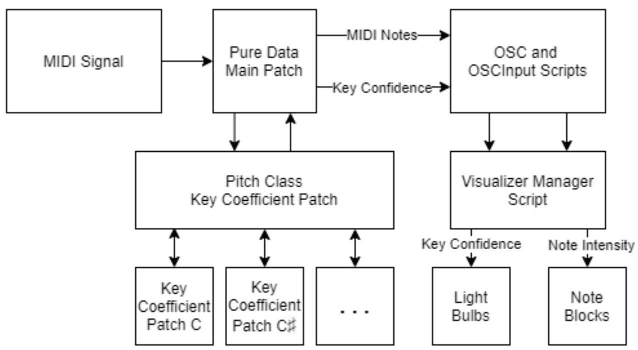

Figure 3: MIDI input flows into our central Pure Data patch, and is then processed to detect key information. Both the individual MIDI signals and key data is passed through OSC into the Unity project where it is used to control the visualization.

The program consists of two major components: a Pure Data patch that interprets MIDI note data as key confidences, and a Unity project that visualizes the note data and key confidences using the virtual reproduction of Scriabin's clavier à lumières. Pure Data is an open source visual programming environment for interactive audio applications developed by Miller Puckette; Pure Data programs, known as patches, are created using a visual interface of objects connected by patch cords. Unity is a general purpose game engine that is an industry standard, making it a flexible and robust platform for this project. To create a line of communication between these two components, we use the Open Sound Control (OSC) protocol to send and receive messages between the two. While MIDI is the protocol for transmitting audio data outfitted in most keyboards, OSC allows for precise labelling and structuring of specific pieces of data, making it ideal for a project using such unique musical data.

Users interface with the visualizer simply by playing notes on a piano keyboard. The program can use any piano keyboard with MIDI functionality as a user interface; when a performer plays the keyboard, the resultant MIDI note data is sent to the Pure Data patch. This data is fed into the Pure Data implementation of the Krumhansl-Schmuckler algorithm, generating the key confidence values. These values are then formatted as OSC messages alongside the initial note data and transmitted to the Unity build. Here, the OSC messages are parsed as input for the visualizer.

5 VISUAL EXPRESSION USING UNITY

5.1 Prototyping

Our first attempt at a MIDI-based music visualizer associated HSV hue values with each key based on its distance around the wheel of fifths. It relied on a single particle system which would generate a particle and trail for any note played. Each particle and trail flowed up from the bottom of the screen, its horizontal position determined by its place on a keyboard. Particles and paths for notes within whatever was the reckoned key would be colored appropriately where notes outside of that key would be colored in the most contrasting hue.

When engaging with Scriabin's system, it became clear that this representation would not be suitable. It collapses musical color into keys ignoring color associations which would be present with individual notes. It also did not take advantage of Unity's many affordances such as the ease of use it provides in managing three-dimensional models, their render materials, and lighting.

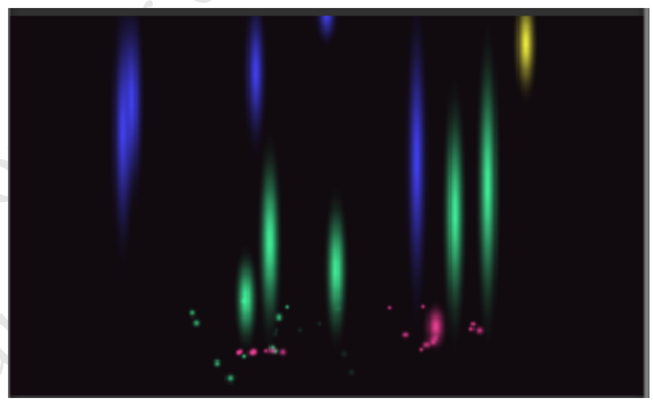

Figure 4: Output of the first pass at a visualizer to interpret melody action and likely key state.

5.2 Visualizer Space

Our final design for the visualizer takes inspiration from the device that Scriabin used in performances of his "Poem of Fire" [1] that is currently present in the Alexander Scriabin Museum in Moscow. The device features twelve colored light bulbs set into fixtures around a circular wooden board. The color of each build is associated with one key around the circle of fifths and its given color within Scriabin's system. A controller allows the user to independently activate any of the bulbs.

In our visualizer we create an analogous board of bulbs that are controlled by the visualizer program. Building within Unity allows us to take advantage of the engine's built-in lighting system. Our Scriabin circle-of-fifths bulbs illuminate the interior of an inverted cylinder. The use of this particular object within our visualizer allows the Unity camera to see into the structure from the outside and capture the entirety of the visualizer space. The interior is colored a dark neutral gray to control the effects of tweaking the ranges and intensity of light objects within the space.

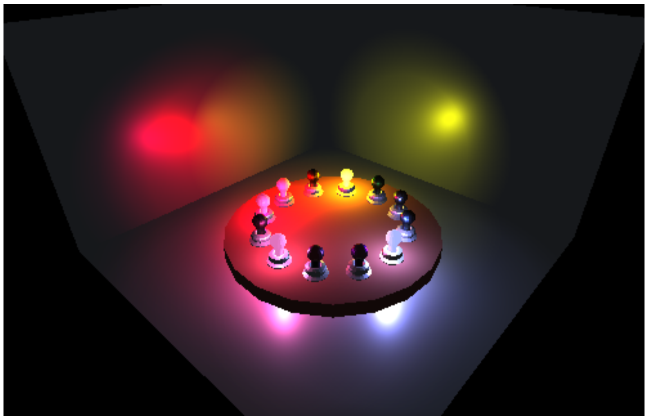

Figure 5: The first prototype of the Scriabin inspired light-bulb circle

6 VISUALIZER MANAGER

The Visualizer Manager script handles the ongoing state of the visualizer. It manages the collection of colored light bulbs and the sixty note blocks. For each possible major key, it stores a confidence value represented by a float. Every time the executable receives a signal of a change in key state, the manager updates the appropriate key and the state of the corresponding light bulb object. If the confidence value is greater than 0.3 then the light is activated or otherwise deactivated. The confidence value is passed into a variable belonging to the light bulb game object which modulates the intensity of the given light. Light bulbs associated with keys with a higher confidence value glow brighter. When a light is activated, its child light object is enabled for the next rendered frame and its child particle system is triggered, which creates a small explosion of color-appropriate sparks. The particles in this system are given a short lifetime and given a color over lifetime gradient which causes them to fade. This allows for dramatic bursts of color with a soft conclusion.

The Visualizer Manager also manages the collection of sixty note block objects which represent the states of sixty middle notes within the MIDI input range. Like the light bulb objects, each can be activated, has a child light which is triggered, and generates a burst of particles. When the manager receives a /midi message, if the message has an intensity above 0, it is activated. On an intensity level of 0, the note block is deactivated.

7 RESULTS

The visualizer does a proficient job at identifying the key when given major triads; as seen in Figure 6 (top), a C major triad results in a bright red aura like the one associated with that key in Scriabin's notes, and if one plays major triads neighboring C major on the circle of fifths in passing, the visualizer will slightly shift its weights to account for the change, but will generally stay centered on red. The visualizer is also competent at displaying the plethora of tonal possibilities when playing material with less of a tonal center; in Figure 6 (bottom), a C diminished chord results in four equally dim lights, displaying that any of them could be the key center, but none with any certainty that it is identified more brightly than the others. This differentiation is important as dissonant chords such as diminished chords function specifically to upset the listener's established idea of the tonal center and allow for modulation to a new key. This demonstrates that the model is approaching consonance and dissonance in classical Western tonal harmony the same way that a listener would.

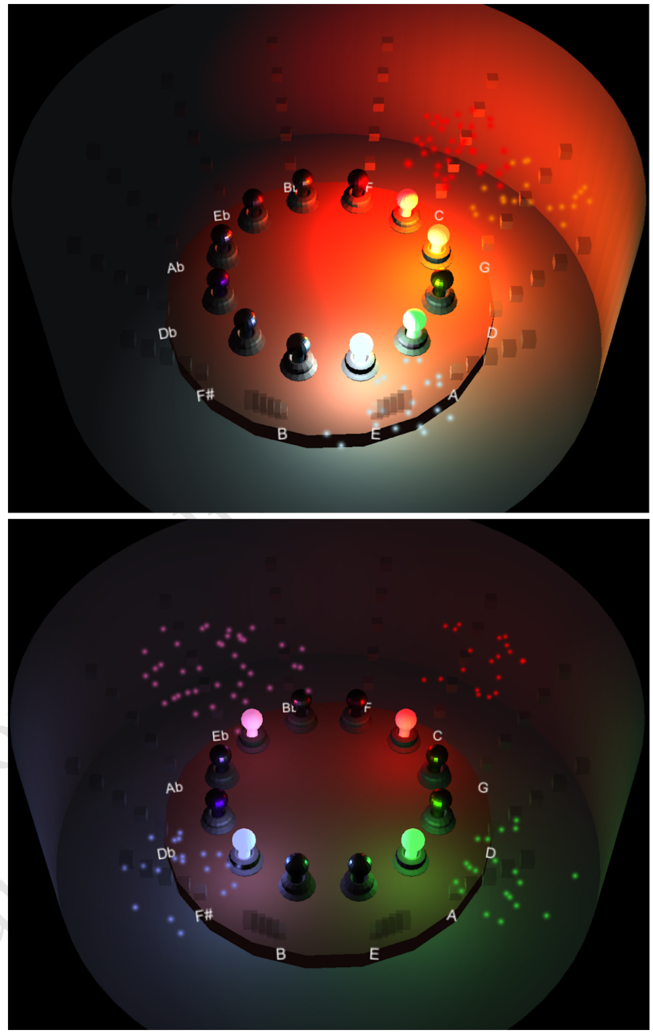

Figure 6: Top: visualizer output using key-detection data from a C major triad. Bottom: output from a C diminished chord. Major chords strongly imply a few keys, leading to a brighter response. Notes making up diminished chords don't characterize any major or minor keys. As such we have a dimmer response.

We do not know all the details of the original experimental study the key profile values were derived from, only that the participants were asked to rate how well each pitch class completed either an ascending or descending scale in C major ending just before the final tonic, and that the participants themselves were apparently university students with varying musical experience. Later experiments included chordal accompaniment to more strongly "frame" the probe tones within a specific key, as well as recruiting participants that were all experienced musicians. The resulting data and analysis are not surprising. The tones which are conventionally understood in the classical musical tradition as stable, turn out to be experimentally supported as stable. But musical acculturation of the participants could explain the experimental results. This is perhaps the reason why it is easier to use our system to visualize tonality in the frame of classical Western tonal harmony, and why the system finds it more difficult to visualize music that draws from musical traditions outside of that cultural context.[2] For example, in order to create a model of tonality better suited to a jazz framework, one would need to base it on an ontology inclusive of triadic extensions in order to determine key.

REFERENCES

[1] Mariya Akimova. 2021. *Alexander Scriabin Museum.* Moscovery. Retrieved March 12, 2021 from https://www.moscovery.com/alexander-scriabin-museum/

[2] Philip A. Ewell. 2020. Music Theory and the White Racial Frame. *Music Theory Online* 26, 2 (Sep 2020). https://doi.org/10.30535/mto.26.2.4

[3] B. M. Galeyev and I. L. Vanechkina. 2001. Was Scriabin a Synesthete? *Leonardo* 34, 4 (Aug 2001), 357–361. https://doi.org/10.1162/00240940152549357

[4] Nick Kanozic. 2018. Light Organ - Scriabin No 1. Video. Retrieved March 17, 2021 from https://www.youtube.com/watch?v=6ZEjZ-UpWoM

[5] Carol L. Krumhansl. 1992. Cognitive Foundations of Musical Pitch. *Music Perception: An Interdisciplinary Journal* 9 (07 1992), 476–492. https://doi.org/10.2307/40285567

[6] Thom Stylinski. 2010. https://www.youtube.com/watch?v=V3B7uQ5K0IU. Video. Retrieved March 17, 2021 from https://www.youtube.com/watch?v=V3B7uQ5K0IU