



УДК 78.02

ИСПОЛЬЗОВАНИЕ СРЕДЫ “MAX/MSP/JITTER” В ТВОРЧЕСКОЙ ПРАКТИКЕ КОМПОЗИТОРОВ XX–XXI ВВ.

А.Ю. МИХЕЛЬ

Новосибирская государственная консерватория имени М.И. Глинки, 630099, Новосибирск, Российская Федерация

АННОТАЦИЯ. В статье рассматривается один из широко используемых инструментов, ставший в последние десятилетия мировым стандартом в области визуального программирования для создания произведений в различных направлениях электроакустической музыки. Сегодня освоение Max/MSP/Jitter является обязательной частью образовательных программ крупнейших центров электроакустической (акусматической) музыки, саунд-дизайна, музыкальной информатики, мультимедиа искусства, исследовательских и экспериментальных студий электронной музыки. Автор обзорно представляет историю развития Max/Msp/Jitter, принципы её организации и способы применения на практике. В работе анализируются особенности Max/Msp/Jitter, позволяющие композиторам и мультимедиа-артистам воплощать смелые художественные замыслы, создавая произведения на основе новых композиционных подходов, таких как стохастические процессы, клеточные автоматы, генетические алгоритмы и т.д., используя для реализации своих идей широкий спектр средств (синтез и обработку звука, работу с видео и изображениями, 3D-графику, физическое моделирование, а также различные устройства, такие как MIDI-контроллеры, сенсоры, системы виртуальной реальности, проекционный маппинг и т.п.). На примере сочинений Пьера Булеза “Anthèmes 2”, Ламберто Кочиолли “Touch” и Карлхайнца Эссла “Lexicon-Sonate” автор анализирует опыт использования композиторами среды Max/Msp/Jitter.

КЛЮЧЕВЫЕ СЛОВА: Max/Msp/Jitter, алгоритмическая композиция, саунд-дизайн, мультимедийная композиция, музыкальное программирование, живая электронная музыка.

КОНФЛИКТ ИНТЕРЕСОВ. Автор заявляет об отсутствии конфликта интересов.

USING THE “MAX/MSP/JITTER” ENVIRONMENT IN THE CREATIVE PRACTICE OF A COMPOSERS OF THE XX–XXI CENTURIES

A.Y. MIHEL

M.I. Glinka Novosibirsk State Conservatory, Novosibirsk, 630099, Russian Federation

ABSTRACT. This article discusses one of the widely used instruments that in recent decades has become the world standard in the field of visual programming for creating works in various directions of electroacoustic music. Today, mastering Max/MSP/Jitter is a mandatory part of the educational programs of the largest centers of electroacoustic (akusmatic) music, sound design, music computer science, multimedia art, research and experimental studios of electronic music. The author provides a history of the development of this environment, the principles of its organization, and ways of applying it in practice. The article analyzes the features of Max/Msp/Jitter that allow musicians and multimedia artists to embody the boldest artistic intentions, creating works based on new compositional approaches such as stochastic processes, cellular automata, genetic algorithms, etc., using a wide range of means to embody their ideas. This includes synthesis and sound processing, working with video and images, 3D graphics, physical modeling, as well as various devices such as MIDI controllers, sensors, virtual reality systems, projection mapping, etc. The article provides examples of the application of Max/Msp/Jitter by composers. The compositions of Pierre Boulez “Anthèmes 2”, Lamberto Coccioli “Touch”, Karlheinz Essl “Lexicon-Sonate” are considered.

KEYWORDS: Max/Msp/Jitter, algorithmic composition, sound design, multimedia composition, music programming, live electronics.

CONFLICT OF INTERESTS. The author declares the absence of conflict of interests.



Max/MSP/Jitter представляет собой комплекс программных инструментов для творчества композиторов и мультимедиа художников. Это комбинированное программное обеспечение позволяет создавать интерактивные музыкальные и видео проекты, используя графический интерфейс, а не кодирование.

Max/MSP/Jitter является визуальным языком программирования, имеющим модульную структуру. MSP (Max Signal Processing) – это расширение Max, добавляющее возможность работы с цифровым аудио. Jitter – ещё одно расширение, позволяющее работать с многомерными массивами данных (матрицами), видео и 3D графикой.

Использование среды Max/MSP/Jitter в творческой практике современных композиторов является сегодня распространённым явлением. Всё больше музыковедов уделяют в своих исследованиях внимание таким областям как электроакустическая, алгоритмическая и мультимедийная композиции, в которых использование такого инструмента как Max/MSP/Jitter играет весомую роль.

Вместе с тем в русскоязычной научной литературе к настоящему времени данная проблематика ещё не получила всестороннего освещения. Это обстоятельство обуславливает актуальность настоящей статьи. Её цель – рассмотрение способов использования Max/MSP/Jitter в создании музыкальных и аудиовизуальных произведений, новых алгоритмических и интерактивных методов композиции, анализ опыта использования Max/MSP/Jitter композиторами в своих произведениях.

Max/MSP/Jitter (далее просто Max) имеет интересную историю развития. Он был создан в 1980-х гг. двумя американскими композиторами Миллером Пакеттом и Дэвидом Зиксом в Институте исследования и координации акустики и музыки (IRCAM) в Париже [8]. Разработчики хотели придумать техническое средство для создания и исполнения музыки, которое было бы полностью интерактивным и контролируемым.

Первую версию Max М. Пакетт и Д. Зикс официально презентовали профессиональному сообществу в 1986 году. Она представляла собой среду разра-

ботки, позволявшую создавать инструменты для сочинения и управления музыкой в реальном времени. В течение следующих лет авторы продолжали разрабатывать Max, добавив новые возможности и функции, такие как MSP и Jitter [6].

Max 2.0, выпущенный в 1990 году, содержал большее количество объектов и функций для обработки звука и MIDI, а также возможность интеграции с другими программами для создания музыки.

В Max 3.0 (1994) было добавлено расширение для работы с аудио – MSP, и объекты для работы с файлами.

Max 4.0 1998 года включал новые объекты для управления последовательностями событий и типами модуляции¹, а также расширенные возможности обработки MIDI.

В Max 4.5 (2002) появилось расширение Jitter для работы с видео и изображениями, а также улучшения в работе с внешними устройствами, такими как MIDI-контроллеры.

Max 5 (2005) содержал новые способы для работы со звуком; был обновлён интерфейс и система синхронизации.

Max 6 (2011) дал возможность повысить производительность, а также разработчики добавили расширение Gen, позволяющее создавать эффективные аудио- и видео процессы, используя код на языке программирования C.

В Max 7 (2014) был усовершенствован интерфейс и система управления проектами, введены новые объекты и функции, расширены возможности работы с графикой.

Max 8 (2018) является последней версией программы на данный момент. В неё добавлены новые объекты для работы с виртуальной реальностью, улучшена производительность, обновлён интерфейс и система документации.

Отметим, что язык программирования Max назван в честь композитора Макса Мэтьюза (1926–2011),

¹ Имеется в виду процесс изменения параметров управляемых (модулируемых) сигналов в зависимости от изменения амплитуды или фазы управляющих (модулирующих) сигналов.

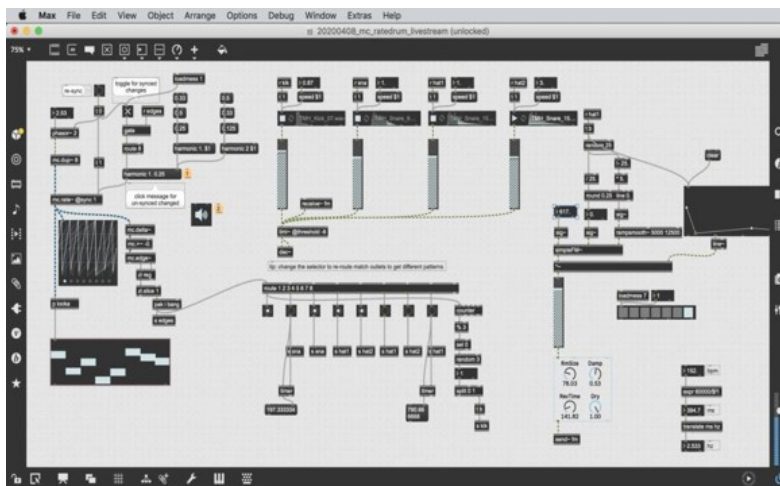


Рисунок 1. Пример патча в Max

который был профессором музыки в Университете штата Калифорния и являлся «пионером» в области создания компьютерных музыкальных инструментов и интерактивных музыкальных произведений. Его работа в этой области включала разработку Music IV и Music 360, предшественников Music V, а также различных других программ, в том числе Patcher, разработанный М. Метьюзом в 1980-х годах в Центре музыки технологии и инноваций в Беркли (CNMAT). Именно благодаря работе М. Метьюза и его команды, на основе идей, заложенных в Music V, но с использованием графического интерфейса пользователя, созданного в процессе работы над проектом Patcher, в дальнейшем был разработан Max.

Max – визуальный язык программирования, основанный на концепции «патчей» (patches) и «патчкордов» (patch cords). Визуальность заключается в том, что программист создаёт программу не путём написания кода, а посредством составления графических схем, где каждый элемент является блоком, называемым «объектом» (object) [6].

Патч – это название для файла проекта в Max, который содержит объекты, соединяемые патчкордами, чтобы образовать графическую программу (рисунок 1). Патчкорды – это соединительные линии между объектами на графической схеме, через которые поток данных передаётся от одного объекта другому.

Объект является программной конструкцией,

представляющей собой элементарную единицу языка программирования Max, предназначен для выполнения различных задач, таких как синтез и обработка звука, работа с видео и 3D-графикой, манипулирование MIDI-сигналами, работа с сетью и других. Объекты могут принимать входные данные, обрабатывать их и генерировать выходные данные, которые затем могут быть направлены на другие объекты или устройства. Объекты имеют свойства, именуемые *аргументами* и *атрибутами*, которые могут быть настроены для изменения их поведения [11].

В Max также доступны текстовые объекты, позволяющие пользователю написать код на других языках программирования, таких как JavaScript и Lua, и к тому же имеется возможность создавать пользовательские объекты с помощью языков программирования C или C++.

Вот некоторые из наиболее распространённых классов объектов в Max:

- 1/ Объекты для работы с числами: float, int, counter, line, metro
- 2/ Объекты для работы со звуком: adc~, dac~, groove~, wave~, buffer~, vline~, biquad~
- 3/ Объекты для управления событиями: button, message, umenu, menu, multislider, panel, pattr, pictslider
- 4/ Объекты для работы с текстом: coll, dict, sprintf, textedit, tosymbol, zl
- 5/ Объекты для работы с видео: jit.qt.movie, jit.



pwindow, jit.matrix

6/ Объекты для работы с 3D графикой: jit.gl.mesh, jit.gl.slab

7/ Объекты для работы с сетью: udpserver, udpreceive, tcpclient, tcpserver, mxj net

8/ Объекты для работы с файлами и файловыми системами: openpanel, savepanel, sfrecord~, sfplay~, coll, dict

9/ Объекты для работы с MIDI: midiin, midiout, notein, noteout, pgmin, pgmout, ctlin, ctout

10/ Объекты для работы с аудиоэффектами: clip~, filtergraph~, biquad~, delay~, meter~, svf~

11/ Объекты для работы с анализом и обработкой звука: fft~, ifft~, fftin~, fftout~, sigmund~, fiddle~, bonk~

Max также содержит встроенный «планировщик событий», позволяющий программисту задавать точное время запуска и длительность выполнения каждого процесса. Это даёт более точный контроль над временными параметрами программы, чем в стандартных языках программирования. Ещё одно отличие Max заключается в том, что код в нём может выполняться параллельно, то есть существует возможность выполнения разных частей программы одновременно. Max имеет функции мультипроцессорного использования, которые позволяют распределять процессы обработки звука на несколько ядер процессора, что обеспечивает высокую производительность при обработке большого количества звуковых потоков [5].

Max является мощным инструментом для создания интерактивной музыки и звукового дизайна, что делает его очень привлекательным для композиторов, звукорежиссёров и звуковых художников. Max обладает огромным потенциалом для синтеза, обработки и анализа звука, а также управления им в реальном времени [12, с. 9]. Композиторы могут использовать эту среду для создания интерактивных инсталляций, мультимедийных произведений и фонограмм для фильмов и видеоигр, а также для создания уникальных инструментов, которыми можно руководить в реальном времени, или для автоматизации различных задач, связанных со звуковым дизайном [10]. С помощью Max композиторы имеют возможность формировать нестандартные интерфейсы для командования мультимедийными процессами, которые могут управляться с помощью

физических контроллеров, таких как MIDI-клавиатуры, гитарные эффекты, сенсоры и любые другие устройства ввода [7, с. 185–199].

Рассмотрим подробнее опыт использования композиторами среды Max/Msp/Jitter на примере произведений Пьера Булеза, Ламберто Кочиолли и Карлхайнца Эсса.

Как известно, Пьер Булез (1925–2016) был одним из первых композиторов, кто начал экспериментировать с электронной музыкой ещё в конце 1940-х гг. Он создал ряд работ, которые считаются монументальными в истории электронной музыки, включая “Répons” (1981), “Dialogue de l’ombre double” (1985). В последующие годы музыкант продолжал исследовать новые звуковые технологии, используя в своих композициях как электронику, так и традиционные инструменты.

П. Булез является основателем центра IRCAM², открытого в Париже в 1977 году для поддержки исследований в области электронной и компьютерной музыки. П. Булез был одним из основоположников развития компьютерных программ для композиторов, и его работа с IRCAM существенно способствовала продвижению новых технологий и направлений в музыке [1, с. 9].

“Anthèmes 2” – это сочинение для скрипки и электроники (1997), которое является продолжением его ранней работы “Anthèmes” для скрипки соло (1991). “Anthèmes 2” является показательным примером использования технологии интерактивной обработки звука в реальном времени³ [8]. Композитор применил Max для обработки и управления звуком скрипки, а также для создания электронной партии.

Патч “Anthèmes 2” был создан таким образом, что система “Live electronics” считывала информацию о производимых на скрипке звуках, а затем обрабатывала эту информацию, изменяя тембр и громкость

² IRCAM-Institut de Recherche et Coordination Acoustique/Musique (Институт исследования и координации акустики и музыки).

³ Произведение состоит из трёх частей и длится около 12-ти минут. В первой части скрипка играет соло, а электроника образует эффект пространственного звучания. Во второй – скрипка и электроника взаимодействуют, создавая наложения звуков посредством эффекта задержки. В третьей части виртуозная скрипичная тема соотносится со звуком, обработанным электроникой.

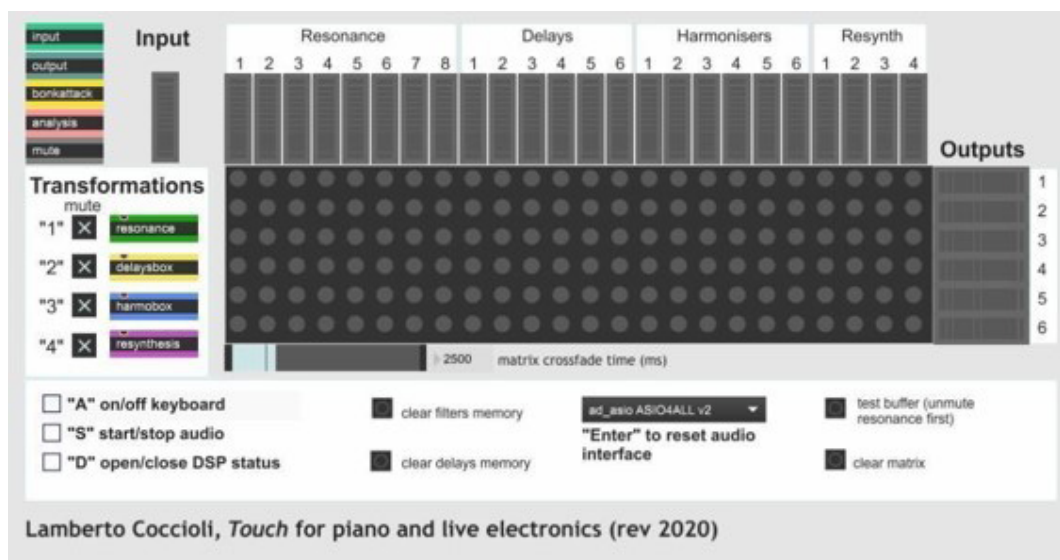


Рисунок 2. Патч "Touch" в Max

звука. В результате программа позволила П. Булезу создать электронные звуки, сложные звуковые эффекты и управлять ими в реальном времени в сочетании с натуральным тембром скрипки [там же]. Таким образом, Мах выступил интегральной частью процесса создания музыкальной композиции "Anthèmes 2" и позволил оригинально расширить границы традиционного звучания инструмента.

Ламберто Кочиолли (род. 1963) – современный итальянский композитор, исполнитель и педагог, активно применяющий Мах в своём творчестве. Он изучал композицию у Эдгара Аланди и Ацио Корги в Милане, посещал летние курсы и мастер-классы у Пьера Булеза, Эллиота Картера и Джорджа Бенджамина⁴. Важное влияние на его творчество оказали путешествия в отдалённые районы Колумбии, где он записывал музыку и звуки традиционных индейских и метисских сообществ [10].

В 2000 году Л. Кочиолли стал заведующим кафедрой музыкальных технологий в Королевском консерватории Бирмингема, где возглавлял проект «Интегра – слияние музыки и технологии», поддерживаемый Европейским союзом для продвижения и исполнения музыки с живой электроникой

(2005–2012), включая разработку программного обеспечения "Integra Live". В 2009 году он совместно с Джейми Буллоком основал центр исследований по дизайну музыкального взаимодействия, базирующийся в Королевской консерватории Бирмингема. С 2012 года является её профессором музыки и технологии [там же].

Композиция "Touch" для фортепиано и live электроники⁵ представляет собой пример эксперимента композитора с целью создания оригинальной исполнительской среды, в которой технологии расширяют музыкальные жесты исполнителя. Четыре одинаковых музыкальных объекта в "Touch" рассматриваются Л. Кочиолли с разных точек зрения, каждый раз с использованием разных инструментов преобразования, чтобы подчеркнуть всевозможные аспекты игры на фортепиано: прикосновение, резонанс, гармоническое и мелодическое измерения. Исполнитель полностью контролирует технологию, а внешнее вмешательство сведено к минимуму. Л. Кочиолли добился этого, создав интерфейс, очень тонко реагирующий на нюансы музыкального испол-

⁴ С 1995 по 1999 год Л. Кочиолли был ассистентом Лучано Бериио.

⁵ Премьера сочинения состоялась в 2002 году в Концертном зале Бирмингемской консерватории. Партия фортепиано – Лора Пинсмейл.



нения. Особое внимание было уделено обнаружению атаки звука фортепиано: прикосновение пианиста запускает каждый раз весь процесс преобразования.

В патче "Touch" (рисунок 2) используются разные типы обработки звука, такие как фильтры задержки, транспонирование (Pitch shift), а также анализаторы частотного спектра и атаки [10].

Карлхайнц ЭSSL (род. 1960) – современный австрийский композитор, исполнитель, импровизатор и саунд-дизайнер. Изучал музыковедение в Венском университете, композицию у Фридриха Черха и Дитера Кауфмана. Работал в Париже в IRCAM, преподавал алгоритмическую музыку в Линце и Вене. Его творчество характеризуется использованием компьютеров для создания и исполнения музыки с живой электроникой. К. ЭSSL известен своими интерактивными и генеративными произведениями, такими как "Lexikon-Sonate" и "Amazing Maze".

О творчестве этого композитора можно сказать, что он работает в разных жанрах и стилях, от камерной и оркестровой до электроакустической и экспериментальной музыки. Его музыка часто имеет концептуальный или философский характер, отражая его интерес к языку, математике и эстетике [2].

К. ЭSSL разрабатывает свои собственные программы и модули для обработки и трансформации звука в реальном времени, а также пишет интерактивные и генеративные произведения, реагирующие на входящие звуковые сигналы или на случайные события. Он интересуется алгоритмическими и генеративными методами композиции, позволяющими создавать музыкальные структуры и формы на основе математических правил и моделей, используя Мах как инструмент для импровизации и создания звуковых ландшафтов.

К. ЭSSL известен в профессиональных кругах созданием обширной библиотеки объектов для Мах под названием *Real Time Composition Library* (RTCL)⁶. Он разработал её в начале 1990-х годов и с тех пор RTCL используется многими композиторами и музыкантами. Библиотека предлагает потрясающую возможность экспериментировать с рядом техник музыкальной композиции, например, таких как се-

рийная техника, и контролируемая случайность и др.

Одной из особенностей RTCL является то, что она позволяет композиторам в реальном времени использовать интерфейс, отображающий звуковую структуру композиции на экране. Это крайне удобно, так как помогает быстро создавать новые музыкальные идеи и прослушивать их в реальном времени. RTCL включает в себя разные категории объектов: Toolbox, Chance, Lists, Harmony, Rhythms, Envelopes, MSP objects, Jitter objects.

RTCL использовалась К. ЭССЛОМ при создании одного из его самых известных сочинений – "Lexikon-Sonate" (1994). Это интерактивное, генеративное произведение для компьютерно-управляемого фортепиано (*дисклавира*), в котором музыка формируется в режиме реального времени в зависимости от взаимодействия исполнителя с компьютерной программой. Композитор вдохновился «Лексикон-романом»⁷ австрийско-словацкого писателя Андреаса Окопенко, являющимся одним из первых *литературных гипертекстов*. «Лексикон-роман» устроен как словарь или энциклопедия: он разбит на статьи, упорядоченные в алфавитном порядке и содержащие гиперссылки.

Композитор решил сотворить музыкальный эквивалент данного сочинения [9]. Это привело его к идее структурировать и классифицировать различные музыкальные элементы, такие как мелодия, аккорды, арпеджио, глиссандо, различные фигуры, мелизмы и т.д., и создать программу, с помощью которой исполнитель может импровизировать, изменяя параметры генераторов стохастических структур, которые создают эти музыкальные элементы в реальном времени. Стоит отметить, что при создании генераторов К. ЭSSL использовал стилистические особенности разных композиторов, определив пути развития музыки (И.С. Баха, В.А. Моцарта, Ф. Шуберта, Й. Брамса, А. Шенберга и др.).

"Lexikon-Sonate" является электроакустической композицией, в основе которой лежит патч Мах (рисунок 3), созданный на основе RTCL и состоящий из 24 различных модулей. Каждый из них отвечает за определённые музыкальные параметры, такие

⁶ RTCL доступна для бесплатного скачивания и использования на авторском сайте К. ЭССЛА (<https://www.essl.at/works/rtc.html>).

⁷ Полное его название – «Лексикон сентиментальной поездки на встречу экспортеров в Друдене» ("Lexikon einer sentimental Reise zum Exporteur-Treffen in Druden", 1970 г.



Рисунок 3. Патч "Lexikon-Sonate" в Max

как ритм, темп, динамика и т.д. Исполнитель может взаимодействовать с программой: выбирать модули и менять их параметры для генерации музыкальных фрагментов. Затем они комбинируются и трансформируются, создавая таким образом уникальную композицию.

Каждое исполнение "Lexikon-Sonate" является непредсказуемым и будет отличаться от предыдущего из-за того, что К. ЕSSL использует более 70-ти разного типа генераторов случайных чисел, которые соединены сложным образом. Произведение не имеет фиксированной нотации или структуры и постоянно меняется в зависимости от входящих параметров. Оно может быть воспроизведено на любом компьютерно-управляемом фортепиано, таком как Yamaha Disklavier или Bösendorfer CEUS. Исполнитель с помощью MIDI контроллера подключается к компьютеру с установленной программой и изменяет параметры. Программа генерирует MIDI сообщения с информацией о высоте ноты и силе нажатия; затем сообщения передаются на дисклавир

и сочинение обретает своё звуковое воплощение. Таким образом можно сказать, что произведение существует как бы в трёх ипостасях: во-первых, это программа; во-вторых, это процесс исполнения; и, в-третьих, это сам звук.

В заключение подчеркнём, что перспективы Max/MSP/Jitter и его применения в области алгоритмической музыки, синтеза и обработки звука, аудиовизуальных инсталляций и других мультимедийных арт-проектов весьма обширны. Это связано с ростом интереса к произведениям, которые задействуют технологии интерактивности. Одним из оригинальных направлений развития Max является расширение возможностей программы для работы с искусственным интеллектом и машинным обучением. Это позволит создавать ещё более сложные и интерактивные произведения, а также использовать Max в более широком круге приложений, которые могут быть использованы в различных областях – театр, кино, телевидение, реклама и игровая индустрия.



ЛИТЕРАТУРА

- 1/ Ковалёв А.В. Случайность: Малларме и Булез // Музыкальная академия. 2019. № 1. С. 6–15.
- 2/ Липов А.Н. Карлхайнц Эssl. Изменения в электроакустической музыке // PHILHARMONICA. International Music Journal. 2016. № 4. С. 42–75.
- 3/ Соколовский Д.В. Проблемы и перспективы развития системы высшего образования в области высокотехнологического звукового искусства // Электронный научно-исследовательский журнал Сибирского государственного института искусств имени Дмитрия Хворостовского «ARTE». 2022. № 4. С. 64–73.
- 4/ Фатьянова Е.А. Симулякры в электронной музыке: имитация акустических тембров // Проблемы музыкальной науки / Music Scholarship. 2021. № 4. С. 161–170. DOI: 10.17674/2782-3601.2022.1.161-170
- 5/ Cipriani A. Electronic Music and Sound Design: Theory and Practice with Max/MSP: Vol. 1 / A. Cipriani. Rome: ConTempoNet, 2013. 538 с.
- 6/ Cycling 74: Официальный сайт. URL: <https://cycling74.com/> (дата обращения 05.05.23)
- 7/ Goudeseune M.C. Composing with parameters for synthetic instruments: diss.; University of Illinois at Urbana-Champaign. Urbana-Champaign, 2001. 268 с.
- 8/ Ircam: Официальный сайт. URL: <https://www.ircam.fr/> (дата обращения 05.05.23).
- 9/ Karlheinz Essl: Официальный сайт. URL: <https://www.essl.at> (дата обращения 05.05.23).
- 10/ Lamberto Caccioli: Официальный сайт. URL: <https://www.lambertocaccioli.com/> (дата обращения 05.05.23).
- 11/ Manzo V.J. Max/MSP/Jitter for Music: A Practical Guide to Developing Interactive Music Systems for Education and More. New York: Oxford University Press, 2016. 432 с.
- 12/ Sweeton M. Development of Compositional Tools in Max/MSP: master's thesis; University of Huddersfield. Huddersfield, 2011. 68 с.

REFERENCES

- 1/ Kovalev, A.V. (2019), "Randomness: Mallarme and Boulez", *Muzykal'naya akademiya [Music Academy]*, No. 1, pp. 6–15. (in Russ.)
- 2/ Lipov, A.N. (2016), "Karlheinz Essl. Changes in Electroacoustic Music", *PHILHARMONICA. International Music Journal*, No. 4, pp. 42–75. (in Russ.)
- 3/ Sokolovsky, D.V. (2022), Problems and prospects for the development of the higher education system in the field of high-tech sound art, *ARTE*, No. 4, pp. 64–73. (in Russ.)

- 4/ Fatianova, E.A. (2021), "Simulacra in Electronic Music: Acoustic Timbres Imitation", *Problemy muzykal'noi nauki / Music Scholarship [The Challenges of Music Science / Music Scholarship]*, No. 4, pp. 161–170. DOI: 10.17674/2782-3601.2022.1.161-170 (in Russ.)
- 5/ Cipriani, A. (2013), *Electronic Music and Sound Design: Theory and Practice with Max/MSP Vol. 1*, ConTempoNet, Rome, 538 p. (in Eng.)
- 6/ Cycling 74: Official website, Available at: <https://cycling74.com/> (Accessed 05 May 2023). (in Eng.)
- 7/ Goudeseune, M.C. (2001), *Composing with parameters for synthetic instruments: diss.*; University of Illinois at Urbana-Champaign, Urbana-Champaign, 268 p. (in Eng.)
- 8/ Ircam: Official website. Available at: <https://www.ircam.fr/> (Accessed 05 May 2023). (in Eng.)
- 9/ Karlheinz Essl: Official website. Available at: (Accessed 05 May 2023). (in Eng.)
- 10/ Lamberto Caccioli: Official website. Available at: <https://www.lambertocaccioli.com/> (Accessed 05 May 2023). (in Eng.)
- 11/ Manzo, V.J. (2016), *Max/MSP/Jitter for Music: A Practical Guide to Developing Interactive Music Systems for Education and More*, Oxford University Press, New York, 432 p. (in Eng.)
- 12/ Sweeton, M. (2011), *Development of Compositional Tools in Max/MSP: master's thesis*; University of Huddersfield, Huddersfield, 68 p. (in Eng.)

Сведения об авторе

Михель Артур Юрьевич, ассистент-стажёр теоретико-композиторского факультета, Новосибирская государственная консерватория имени М. И. Глинки (научный руководитель – доктор искусствоведения, профессор Л.В. Александрова)
E-mail: arturmihel@yandex.ru

Author information

Artur Yu. Michel, trainee assistant of the Theoretical and Composer Faculty, M.I. Glinka Novosibirsk State Conservatory (Scientific Supervisor – D. Sc. (Art Criticism), Professor Lyudmila V. Alexandrova)
E-mail: arturmihel@yandex.ru