

**UNIwersYTET JANA KOCHANOWSKIEGO W KIELCACH
WYDZIAŁ PEDAGOGICZNY I ARTYSTYCZNY**

Dziedzina
SZTUKI PLASTYCZNE

Dyscyplina
SZTUKI PIĘKNE

nr. albumu: 110156
mgr JAKUB MATYS

ALGORYTM, WZÓR, PROCEDURA. OBRAZY HYBRYDOWE

Rozprawa doktorska napisana pod kierunkiem
prof. zw. dr hab. Wiesław Łuczaj

Kielce 2018

SPIS TREŚCI:

WSTĘP	4
PREKURSORY, TWÓRCY, INSPIRACJE	6
LOSOWOŚĆ	8
ALGORYTM	14
WZÓR	19
PROCEDURA	21
OPIS PRAC I ZASTOSOWANYCH TECHNOLOGII	23
KONKLUZJA	29
STRESZCZENIE	30
REPRODUKCJE PRAC	31
ENGLISH DESCRIPTION OF THE DOCTORAL DISSERTATION	44
ZAŁĄCZNIKI	69
BIBLIOGRAFIA	72

WSTĘP

Zdaję sobie sprawę z tego, że opis mojej pracy doktorskiej może odbiegać od standardowych form omawiania oraz interpretowania ekspresji i twórczości artystycznej. Więcej tu będzie objaśnień kwestii dotyczących technologii komputerowych i elektronicznych, zagadnień ze świata fizyki, informatyki czy matematyki, niż ze świata sztuki czy odniesień do tradycyjnych technik malarskich. Jednak nie sposób byłoby wyjaśnić założeń artystycznych i konceptualnych mojej pracy bez choćby pobieżnego omówienia mechanizmów i technologii, których używam, gdyż są one w istocie medium, którym operuję w swojej działalności twórczej.

Nie odnoszę się do myślicieli z dziedzin socjologii, psychologii czy filozofii, ani nawet cybernetyki społecznej. Nie przedstawiam również aspektów socjologicznych moich dzieł, ponieważ nie komentuję kwestii społecznych. Nie przetwarzam graficznie realnego obrazu rzeczywistości. Poszukuję bogactwa graficznego kolejnych iteracji działania wzoru generującego proceduralny obraz w granicach założonego algorytmu, będąc autorem tegoż algorytmu i tej procedury. Moja pracownia to laboratorium, a praca nad dziełem ma charakter eksperymentu cechującego się dużą liczbą powtórzeń procedury generującej obraz. Przedmiotem badania i obserwacji jest działanie algorytmu komputerowego o wysokim stopniu nieprzewidywalności, generującego obraz graficzny oraz ekspozycja tegoż obrazu za pomocą określonych urządzeń wyjścia – ekranu telewizora, rzutnika komputerowego czy, w ostatecznej formie, maszyny malującej wzór z użyciem tradycyjnych narzędzi malarskich. Jakkolwiek użycie maszyny w tworzeniu obrazu malarskiego może budzić skojarzenia z linią montażową Forda w kontekście sztuki i szeroko rozumianego pojęcia makdonaldyzacji George'a Ritzera, jednak i to nie jest założeniem mojej pracy.

W swojej pracy chciałbym przede wszystkim przybliżyć pojęcia związane z medium, którym się posługuję – przede wszystkim losowości, algorytmu, kodu komputerowego i proceduralności w odniesieniu do działalności artystycznej. Eksploracja znaczenia tych zagadnień pozwoli mi w dalszej części za ich pomocą opisać założenia konceptualne mojej twórczości i idee, które wpłynęły na moje postrzeganie sztuki.

O ile mogę zdecydować o niepodejmowaniu rozważań natury filozoficznej czy socjologicznej, bezsprzecznie należy choć wspomnieć o działaniach o podobnej naturze do moich własnych w dziedzinie sztuki ze względu na użyte medium, założenia czy sferę konceptualną, a w szczególności zarysować ich początki. Zastosowanie pierwszych technik komputerowych w działaniach artystycznych czy graficznych dawno już ma za sobą półwiecze; dziesiątki, jeśli nie setki uznanych artystów w swej dziedzinie i pewnie setki tysięcy hobbystów i pasjonatów.

Stosunkowo łatwo wyśledzić początki sztuki komputerowej, jako że są ściśle związane z rozwojem jej głównego medium – elektroniki analogowej, cyfrowej i elektronicznych maszyn obliczeniowych. Zdecydowana większość źródeł donoszących o początkach grafiki komputerowej jako pierwszy podaje obraz wyświetlony przez komputer SAGE przeznaczony do kontroli amerykańskiego systemu radarowego we wczesnych latach pięćdziesiątych. Naukowcy i programiści sprawujący pieczę nad tym komputerem zaprogramowali za pomocą kart perfo-

rowanych swojego podopiecznego tak, aby na prymitywnym (dla nas) monitorze za pomocą linii wektorowych można było wyświetlić kopię bardzo popularnego wówczas rysunku George Petty'ego – portretu jednej z kalendarzowych, roznegliżowanych Pin Up girls.

Mimo że z pewnością nie był to pierwszy obraz wygenerowany i wyświetlony elektronicznie, z dużym prawdopodobieństwem był to jeden z pierwszych, jeśli nie pierwszy, obraz grafiki komputerowej przedstawieniowej prezentujący z dużą dokładnością ludzkie ciało. Widzę w tym przykładzie coś niezwykłego, jakąś formę przepowiedni z początku dziejów maszyn obliczeniowych dla przyszłych pokoleń: oto technologia warta dziesiątki milionów dolarów, powstała aby chronić ludzkość przed możliwą zagładą nuklearną, zamienia się w dłoniach programisty w nośnik popkulturowych treści erotycznych, nasuwając skojarzenia w odniesieniu do możliwych dzisiaj zastosowań tak zaawansowanych technologii, jak sieci internetowe a ich faktycznej zawartości.

Ten przykład, mimo iż wywodzi się z rozważań socjologicznych, których raczej unikam, nasuwa jedną ważną konkluzję – przez długi czas technologie komputerowe pozostawały w rękach naukowców. Z powodu braku przystępnego dla użytkownika interfejsu, urządzeń wejścia wyjścia, moc obliczeniowa komputera zarezerwowana była dla tych, którzy rozumieją jego język – język kodu i algorytmu. Dopiero późniejszy rozwój badań nad interfejsem użytkownika rozpoczęty w latach 70 w laboratorium Xerox Corporation's Palo Alto Research Center (PARC) przyniósł rozwiązania ułatwiające pracę z komputerem przeciętnemu odbiorcy. W tym samym okresie widzimy rozwój i upowszechnienie się komputera osobistego aż do roli jednego z elementów wyposażenia mieszkania typowego konsumenta. Zastosowanie we wczesnych komputerach ośmiobitowych uproszczonej wersji języków programowania dla początkujących, t.j. BASIC, pozwoliło przejąć kontrolę nad mocą obliczeniową tych maszyn entuzjastom i hobbystom którzy nie musieli już poznawać skomplikowanego języka maszynowego i mogli wydawać polecenia komputerom w prostej postaci komend anglojęzycznych i obserwować na ekranie monitorów, i znacznie częściej telewizorów, graficzne efekty zastosowania kodu.

Do dziś pamiętam siebie jako około ośmioletniego chłopca i pierwsze linie kodu przepisane na moim pierwszym komputerze, Atari, z niemieckiego podręcznika do programowania, z którego nie rozumiałem ani słowa. Nie potrafię opisać zdumienia, które odczułem, gdy po uruchomieniu wspomnianego programu na ekranie telewizora moim oczom ukazała się najprawdziwsza błyskawica! Dziś wiem, że była to zwykła łamana biała linia na czarnym tle złożona z wyraźnie widocznych pikseli. Jednak wówczas była to dla mnie prawdziwa błyskawica, siła natury, nad którą zapanowałem za pomocą systemu znaków i symboli alfanumerycznych. W dodatku, po każdym uruchomieniu błyskawica wyglądała całkiem inaczej i pojawiała się w innej części ekranu, a to już był efekt magiczny, którego wówczas jeszcze nie potrafiłem wytłumaczyć i musiałem znaleźć odpowiedź zgłębiając zasady generowania przez komputery liczb losowych. Zanim jednak osoby mojego pokroju mogły usiąść do klawiatur i tworzyć pierwsze generowane komputerowo wzory, znacznie wcześniej inne osoby musiały stworzyć wzory matematyczne definiujące graficzne formuły i wyświetlające je na prymitywnych ekranach oscyloskopów przy pomocy elektroniki analogowej. Już wtedy ich prace rozpoznawano jako sztukę lub, co najmniej, załączek sztuki komputerowej.

PREKURSORY, TWÓRCY, INSPIRACJE

Jako jednego z pierwszych twórcę obrazów generowanych elektronicznie należy wymienić Bena Laposky'ego, który eksponował swoje prace w formie projekcji oscylograficznych przenoszonych na papier światłoczuły pod nazwą "Oscilons". Jego poszukiwania w dziedzinie automatycznie sterowanego abstrakcyjnego wzoru organicznego, które rozpoczął jeszcze w późnych latach czterdziestych, ewoluowały do formy artystycznej przedstawionej na indywidualnej wystawie w Sanford Museum in Cherokee, Iowa w 1952 r. pod tytułem "Electronic Abstractions".

Kolejną ważną postacią w początkach sztuki komputerowej był Austriacki naukowiec Herbert W. Franke, m. in. fizyk, ale i pisarz fantastycznonaukowy oraz artysta. On również wykorzystywał w swoich pracach artystycznych elektroniczny obraz oscylograficzny i technologie fotograficzne do jego utrwalenia. W dodatku Franke pozostawił po sobie ważne dla sztuki komputerowej publikacje jak książka *Computer Graphics: Computer Art* z 1971 czy eseje: *The New Visual language: The Influence of Computer Graphics on Art and Society* oraz *The Expanding Medium: The Future of Computer Art*, w którym trafnie przewiduje dalszy rozwój animacji komputerowej generowanej w czasie rzeczywistym. Mimo istotnej roli dla rozwoju i propagowania sztuki komputerowej, jaką odegrali wyżej wymienieni twórcy, jako jednego z najważniejszych myślicieli wśród nich należy uznać Roya Ascotta, tworzącego od najwcześniejszych lat 60. Jego przemyślenia w dziedzinie sztuki komputerowej, a w szczególności komputerowej sztuki interaktywnej, wydają się być przełomowe. W kontekście mojej pracy najbardziej inspirujące są dla mnie jego dywagacje na temat mediów elektronicznych i komputerowych jako narzędzi, z tekstu *Behaviourist Art and the Cybernetic Vision*. Ascott nie postrzega tu komputera w roli mechanizmu przyrządu podobnego do "narzędzi odlewniczych czy spawarek będących rozszerzeniem siły fizycznej". Widzi on w komputerze coś na kształt instrumentu powiększającego potencjał myśli i wzmacniacz inteligencji, a w moim rozumieniu również idealny środek do przenoszenia idei i formy konceptualnej.

Bezspornie najważniejszym okresem dla początków sztuki komputerowej i wszystkich jej późniejszych odmian były lata 60te a w szczególności rok 1965 w którym odbyły się aż trzy bardzo ważne dla tego nurtu wystawy. Pierwsza to wystawa Georga Neesa *Computer Graphik* na Uniwersytecie w Stuttgarcie. Nees studiujący m.in matematykę i fizykę brał udział w pracach nad konstrukcją maszyn plotujących i doświadczenie to wykorzystywał w swoich pracach artystycznych programując i wykorzystując maszyny rysujące jak i wzory matematyczne do generacji wzorów geometrycznych. Druga to *Computer-Generated Pictures* Michaela Nolla i Beli Julesza w Howard Wise Gallery w Nowym Yorku. Trzecia – *Computer-Grafik (Nake & Nees)* w Galerie Wendelin Niedlich w Stuttgarcie, w której udział wzięli Frieder Nake i, ponownie tego roku, Georg Nees. Wystawa prezentowała prace wykonane przez maszyny plotujące rysunek, sterowane oprogramowaniem z komputera.

Te wydarzenia artystyczne i powszechne uznanie jakie otrzymały ze strony krytyki, to z jednej strony bardzo istotne otwarcie świata sztuki na technologie komputerowe, ale, co najważ-

niejsze w kontekście mojej pracy, stanowiły one otwarcie świata sztuki na dzieła wykonane właśnie za pomocą maszyn plotujących, maszyn CNC – rysujących, malujących, a nawet rzeźbiących. Oczywiście dalsza historia sztuki komputerowej, do dziś włącznie, wskazuje na liczne przykłady artystów poszerzających nurt dzieł sztuki tworzonych z pomocą oprogramowania maszyn czy inteligencji obliczeniowej jak np. twórczość Harolda Cohena, twórcę autorskiego programu komputerowego AARON, tworzącego obrazy autonomicznie zarówno za pomocą ekranu jak i maszyny malującej, czy Pindara Van Armana, który uczynił z budowy robotów malujących swój język wypowiedzi artystycznej, tworząc bodaj bardziej skomplikowany od AARONA system komputerowy Cloudpainter.

Próbując usystematyzować czy choćby przywołać wydarzenia i zjawiska związane z zagadnieniem sztuki komputerowej, łatwo dostrzeżemy jeszcze wcześniejsze działania, przed 1965 r., obejmujące rysunek za pośrednictwem maszyn w rzeźbach kinetycznych Tinguelliego czy w maszynach rysujących Desmonda Paula Henryego, napędzanych również kinetycznie, ale także za pomocą elektroniki analogowej, eksponowanych już na wystawie Ideographs w 1962 r. w Londynie.

Wymienienie wszystkich podobnych przykładów tej dziedziny sztuki przerasta potrzebę i możliwości objętościowe mojej rozprawy. Pragnę jedynie zaznaczyć własną świadomość istnienia poprzedzających moje prace pionierskich, jak i współczesnych działań w tej dziedzinie i ich inspirującego wkładu w moją twórczość.

LOSOWOŚĆ

Obserwując rzeczywistość nie dostrzegam przypadku. Za danym położeniem każdego atomu we wszechświecie stoją niezliczone warunki i siły, które zdecydowały o kształcie znanego nam świata. Ich układ nigdy nie był losowy – wszelkie składowe materii przeorganizowują swoje położenie w nieskończonym łańcuchu przyczynowo skutkowym, niby kule bilardowe uderzone wielkim wybuchem. To, co możemy postrzegać jako losowość jest faktycznie działaniem zmiennych o wysokim stopniu nieprzewidywalności. Mając odpowiednią liczbę danych i wystarczająco wydajny system obliczeniowy moglibyśmy odtworzyć drogę każdego ziarenka piasku, aż do jego położenia na plaży pod naszymi nogami: od procesów naniesienia przez nurt wody i siłę wiatru, a także poprzez uprzednie procesy erozji skał, czy, jeszcze wcześniejsze, procesy terraformacyjne. Choć efekt wydaje się mieć charakter losowy, czy nawet chaotyczny, przyczyny działają zawsze według ściśle określonych zasad istnienia materii. Losowość od dawna inspirowała ludzkość. Historia opowiada o śmiałkach, którzy potrafili oddać swoje istnienie we władanie losu kręcąc bębniem rewolweru z kulą w jednej komorze; byli tacy władcy narodów, którzy nierzadko podejmowali decyzje dotyczące swoich poddanych na podstawie rzuconych kości, kamieni czy listewek z wróżbami tłumaczonymi przez “wybrańców” odczytujących i interpretujących język wszechświata. W grach decydowała o “losie” grających; ich doli, szczęściu, a nawet życiu. Nic więc dziwnego, że i dzisiaj wiele osób wciąż ulega swoistemu urokowi losowości. To co nieprzewidywalne jest boskie, nadprzyrodzone. Powtarzająca się animacja znikającego i pojawiającego się kursora w edytorze tekstu jest w swej monotonii przewidywalna i nikogo nie intryguje obserwacja owego kursora, podróżującego pomiędzy pojawiającymi się literami, zapełniającymi kolejne strony dokumentu. Podobnie w nikim nie wzbudzi zachwyty, szczególnie w kontekście dzisiejszego rozwoju technologicznego, możliwość przemieszczania kwadratu po płaszczyźnie ekranu komputera za pomocą klawiszy klawiatury. Zaintryguje nas jednak, generowana losowo, przez komputer, animacja wspomnianego kwadratu w płaszczyźnie graficznej ekranu połączona z jego multiplikacją i dająca nam wrażenie obserwacji układów nieprzewidywalnych – symulacji życia, swoistej reprezentacji mrówczej farmy rządzonej “sztuczną inteligencją”. Co by było gdybyśmy mogli te ruchy przewidzieć? Czy wówczas nadal doszukiwalibyśmy się w ich cyrkulacji działania losu? Otóż możemy je przewidzieć! Wielu artystów posługuje się “losowymi liczbami” generowanymi komputerowo nie zdając sobie sprawy, iż nie są one rzeczywiście losowe i nie bez przyczyny nazywane są liczbami pseudolosowymi przez informatyków. Wyobraźmy sobie prosty algorytm matematyczny przekształcający jedną dowolną liczbę w inną, w zakresie od 0 do 9. W istocie wystarczy nam wyliczenie cosinusa dowolnej liczby, a następnie jego wartości bezwzględnej, co da nam wartości ułamkowe w zakresie od 0 do 1. Następnie możemy taką liczbę zaokrąglić do jednego miejsca po przecinku i pomnożyć przez 10. W ten sposób uzyskamy wartości mieszczące się w żądanym zakresie. Podając “z głowy” dowolną liczbę w większości przypadków nie będziemy w stanie przewidzieć efektu działania tego wzoru. Najistotniejsze w tym zobrazowaniu jednak jest to, że mamy stuprocentową pewność, że podanie ponownie

tej samej liczby skutkuje wygenerowaniem identycznego wyniku!

Opiszę ten przykład za pomocą pseudokodu – uproszczonej wersji kodu komputerowego – który powinien w klarowny sposób zobrazować działanie algorytmu generowania liczb losowych za pomocą komputera.

Obliczanie wartości L (losowej) dla dowolnej liczby X

1. Weź wartość X
// obliczenie kosinusa liczby X i jego wartości bezwzględnej
2. Jeśli $\cos(X) < 0$ to $L = \cos(X) * -1$
3. Jeśli $\cos(X) \geq 0$ to $L = \cos(X)$
// zaokrąglenie liczby L do jednego miejsca po przecinku
4. $L = \text{round}(L)$
// sprowadzenie liczby L do wartości od 0 do 10
5. $L = L * 10$

Jest to bardzo uproszczony przykład generatora liczb pseudolosowych, w rzeczywistości zaawansowany algorytm systemowy komputera może generować liczby o znacznie większym stopniu nieprzewidywalności oraz spełniające zasady rachunku prawdopodobieństwa, jednak jedna zasada jest niezmienna:

Ta sama liczba podstawiona w miejsce X wygeneruję zawsze tę samą liczbę L.

Jeśli do tak przedstawionego wzoru w miejsce X, w dowolnym momencie decyzyjnym, podstawimy czas zegara systemowego komputera (czyli sprowadzoną do liczby sekund aktualną godzinę i minuty z uwzględnieniem daty) uzyskamy bardzo wysoki stopień nieprzewidywalności wyniku (dodatkowo wartości wejściowe nie będą nigdy powtarzalne w kolejnych losowaniach, jako że zegar komputera się nie cofa). Jednak wartości wygenerowane w dokładnie tym samym czasie na dwóch różnych komputerach będą identyczne.

Z tak przeprowadzonego rozważania wyłania się proporcjonalna zależność między przewidywalnością a losowością: im bardziej zjawisko jest przewidywalne, tym więcej kroków łańcucha przyczynowo skutkowego jesteśmy w stanie odtworzyć wstecz, i tym mniej “tajemniczy” – czy też losowy – wydaje się nam efekt. W generowaniu liczb pseudolosowych liczbę, na podstawie której generujemy wartość pseudolosową, nazywamy ziarnem. W podanym przykładzie ziarnem staje się właśnie liczba sekund zegara systemowego. To pojęcie będzie ważnym elementem dalszych rozważań na temat losowości i algorytmów generujących postać dzieła artystycznego.

Wyższą złożoność działania i wyższy stopień nieprzewidywalności możemy uzyskać na dwa sposoby. Po pierwsze, możemy znaleźć lepsze od zegara systemowego źródło liczb wejściowych (ziaren) czyli znaleźć lub zbudować własny generator ziarna. Po drugie, algorytm sterujący efektem końcowym może posiadać własny zestaw instrukcji reagujący w skomplikowany sposób w zależności od uzyskanego wyniku losowania.

Przykład:

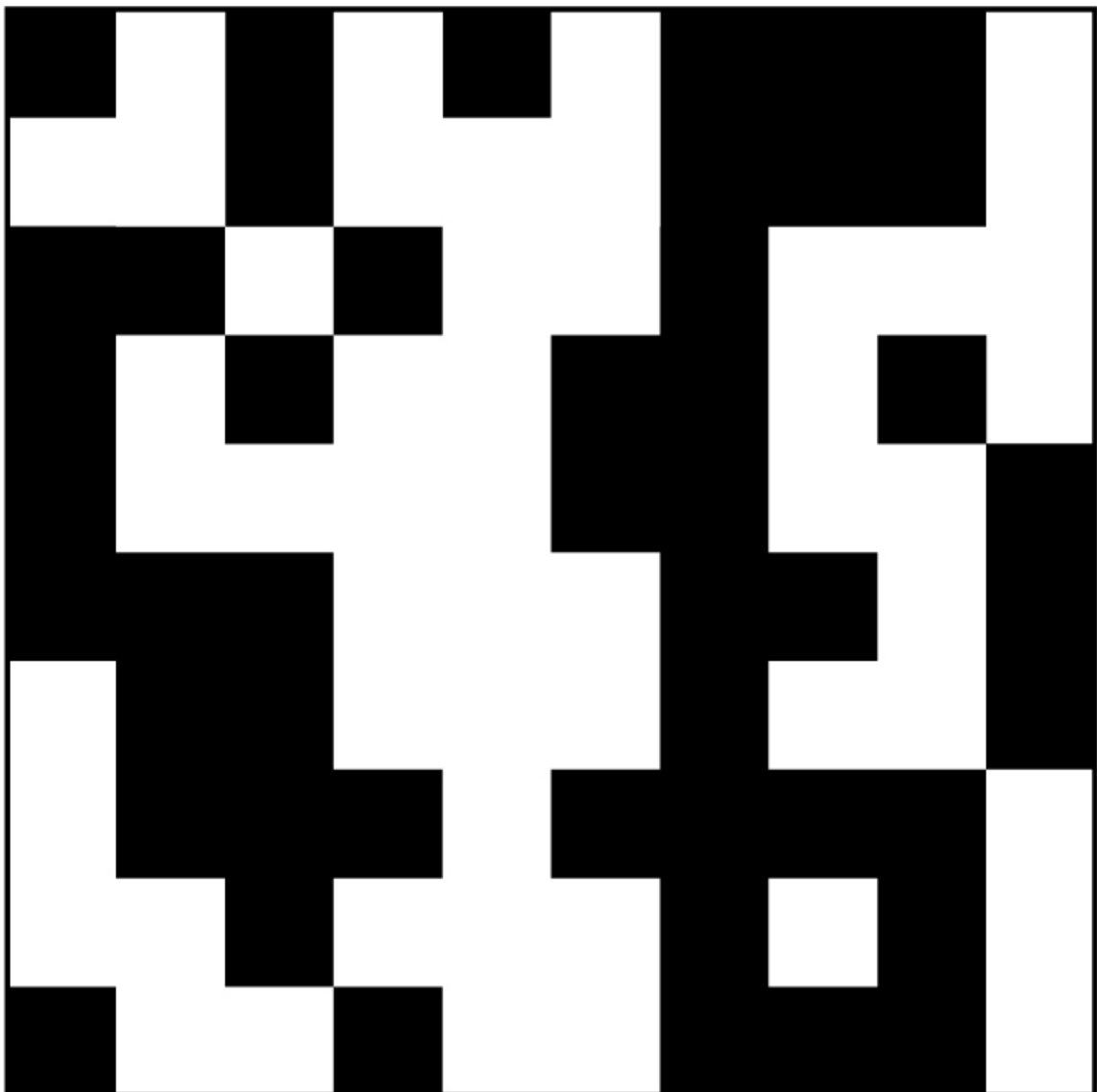
Artysta tworzy obraz czarno-biały, kwadratowy.

Płótno (białe) podzielono na 100 równych pól kwadratowych tworzących grid (siatkę) – 10x10 kwadratów.

1. Artysta rzuca monetą.
2. Jeśli wypadnie orzeł, pole zamalowane zostanie na czarno.
3. Jeśli wypadnie reszka, pole pozostanie białe.
4. Artysta przechodzi do następnego pola i rzuca monetą
5. Powrót do punktu 1.

Tak wykonany algorytm po stu powtórzeniach wypełni wszystkie pola, linijka po linijce, czarnymi i białymi kwadratami, tworząc zapis losowo generowanych wartości.

Jakkolwiek bogaty nie wydawał by się zestaw wariacji tak wylosowanych obrazów, to bez powtórzeń da nam on zaledwie 10 000 różnych efektów końcowych! Choć liczba ta zapewne przekracza ilościowo dokonania życia niejednego artysty, to efekt formalny może przytłaczać swoją monotonią. W dodatku, w zgodzie z zasadami prawdopodobieństwa, wypełnienie całej linijki czernią lub bielą czy uzyskanie przewagi któregoś z kolorów jest wysoce nieprawdopodobne. Co więcej, na większość wyników w jednej linijce (czyli na 10 rzutów) składa się zbliżona ilość orłów i reszek, choć w różnej kolejności. Ta zasada sprawi, że obrazy wygenerowane w ten sposób będą miały graficznie bardzo zbliżony charakter, a tym samym będą dosyć przewidywalne. Oto przykład takiego eksperymentu.



Dochodzimy do punktu, w którym zapewne chcielibyśmy zwiększyć stopień nieprzewidywalności generowanego obrazu i uzyskać ciekawsze graficzne efekty i, jak wspomniałem wcześniej, można spróbować zrobić to na dwa sposoby:

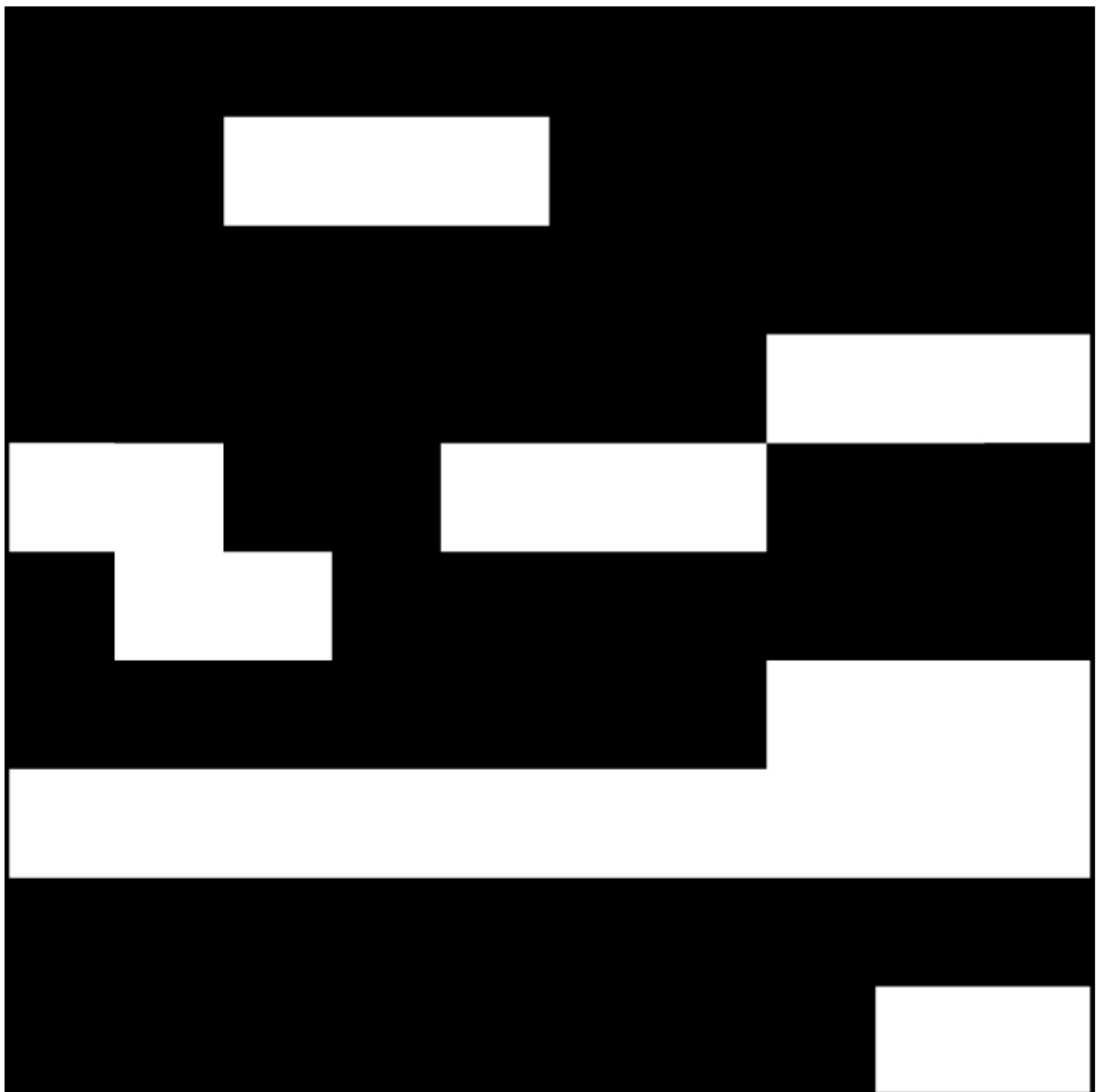
1. Zmieniając metodę generowania wartości losowej
2. Zmieniając zasady wypełniania pól na podstawie wygenerowanej liczby

Nowy algorytm losowania:

Płótno (białe) podzielono na 100 równych pól kwadratowych tworzących grid (siatkę) – 10x10 kwadratów.

1. Artysta rzuca kością sześciocianową.
2. Jeśli wynik jest parzysty na czarno zostanie zamalowanych tyle pól, ile wynosi wynik.
3. Jeśli wynik jest nieparzysty – tyle pól ile wynosi wynik pozostanie białymi.
4. Artysta przechodzi do następnego pola.
5. Powrót do punktu 1.

(rzuty kością : 4,6,2,3,4,4,4,2,6,2,5,2,3,4,1,1,4,6,2,2,3,5,5,4,2,6,2,4,3)



W ten sposób uzyskaliśmy złożoność struktury, której nie moglibyśmy osiągnąć wcześniej uzyskując całkiem inny charakter graficzny obrazu z wyraźną przewagą czerni spowodowaną większymi numerycznie wartościami liczb parzystych. Poszczególne obrazy stworzone za pomocą podobnego algorytmu nie będą identyczne. Będą miały podobny charakter, jednak ich ostateczny kształt będzie zawsze w pewnym stopniu nieprzewidywalny. Możemy na podstawie przytoczonych przykładów zauważyć, że zarówno mechanizm losowania, jak i algorytm decyzyjny kształtujący dzieło wpływają wyraźnie na jego postać. Niezależnie jednak od metody losowania algorytm będzie miał decydujący wpływ na postać graficzną dzieła, tworząc pewnego rodzaju formę, zestaw zasad i ograniczeń, dopuszczając określony zakres losowości. Analizując proces generowania liczb pseudolosowych za pomocą komputera łatwo dostrzec jego słabości dotyczące stopnia nieprzewidywalności. Znanych jest również wiele sposobów lasowania ze świata rzeczywistego, jak wspomniane rzuty monetą, czy rzut kostką sześciocieczną. Z jednej strony procesy te trudno wykorzystać we współpracy z komputerem, z drugiej strony faktycznym elementem losowym jest w nich czynnik ludzki i, w pewnym sensie, też materiałowy. Losowość rzutu kością wynika z siły rzutu, dystansu do podłoża, sprężystości materiału itp. W idealnych warunkach mogąc kontrolować wszystkie te składowe z pewnością moglibyśmy uzyskać wysoką powtarzalność wyniku. Złożoność warunków losowania będzie wpływać na stopień przewidywalności generowanej wartości. Pytanie, w jaki sposób możemy podobny stopień złożoności uzyskać za pomocą mediów cyfrowych? Poszukując rozwiązania dla tej kwestii oraz przyglądając się różnym mechanizmom losowym i ich zastosowaniom, dotarłem do pojęcia szumu.

Szum w świecie technologii cyfrowych i układów elektronicznych jest powszechnym zjawiskiem, z reguły niepożądanym, wprowadzającym niechciane wartości. Pojawia się i jest z uporem zwalczany w technologiach audio operujących sygnałem dźwiękowym i wzmacniających dźwięk. Na pewno wielu z nas pamięta erę telewizji analogowej i charakterystyczny obraz szumu w odcieniach szarości na niedostrojonym do wymaganej częstotliwości kanale. Szum możemy również usłyszeć w niewyregulowanym radiu i w każdej formie przesyłu sygnału audio drogą radiową. Ruch elektronów w fali elektromagnetycznej – jeśli nie zostanie uformowany w odpowiednią amplitudę – jest niespotykane nieprzewidywalny. W dodatku wchodzi tu w grę zakłócenia powodowane przez każde urządzenie elektryczne tworzące swoje własne promieniowanie elektromagnetyczne oraz częstotliwości zawierające szum promieniowania kosmicznego, będącego w dużej mierze pozostałością po wielkim wybuchu. Wszystkie te dane o charakterze chaotycznym możemy wykorzystać jako źródło sygnału losowego o nieporównywalnym z niczym stopniu nieprzewidywalności. Dane te, to po prostu nieprzewidywalne skoki napięcia, które w prosty sposób możemy zamienić za pośrednictwem komputera w dane liczbowe o dowolnym zakresie. Kolejnym źródłem szumu jest ruch elektronów w elektronicznych elementach półprzewodnikowych znany pod nazwą "efektu śrutowego". Efekt ten, choć z reguły niepożądany, można z premedytacją wywołać uzyskując tzw. generator szumu białego (często wykorzystywany przez fachowców w systemach nagłaśniających do testowania zakresu przenoszonego przez głośniki dźwięku). Dostępność tej technologii, jak i możliwość dostosowania schematu konstrukcji do własnych potrzeb, zdecydowały o tym, że tej konkretnej

technologię postanowiłem użyć do realizacji własnych prac, wykorzystując nieprzewidywalny ruch elektronów w tworzenia dzieła artystycznego o charakterze proceduralno losowym.

Dodatkowym czynnikiem decydującym o użyciu właśnie tej technologii był fakt niepowtarzalności stawiający ją ponad zastosowaniem liczb pseudolosowych generowanych przez komputer czy szumu wydobytego za pomocą odbiornika radiowego. Jak wspominałem wcześniej, dwa różne komputery generujące liczbę pseudolosową w dokładnie tym samym czasie wygenerują tę samą wartość. Z kolei szum radiowy w określonym czasie i miejscu to sygnał importowany do odbiornika z zewnątrz i sam odbiornik nie jest tu generatorem, a jedynie odbiorcą zmiennych losowych. Podobnie jak w przykładzie z komputerami – dwa odbiorniki radiowe w tym samym miejscu i czasie otrzymają dokładnie ten sam przekaz, a dwa generatory szumu wykonane z identycznych fabrycznie części, niezależnie od czasu i miejsca, nigdy nie wygenerują tych samych wartości. Również konceptualnie odpowiadał mi fakt, że sam buduję urządzenie, w którym proces generowania zmiennych losowych bierze swój początek.

Można powiedzieć że finalnie nikt nie dostrzeże różnicy między dziełem, w którym liczby będą generowane prawdziwym rzutem kości, a tym, w którym liczby zostały wygenerowane za pomocą komputera i algorytmu pseudolosowego. I jest to na ogół prawda. Komputerowe algorytmy są tak skonstruowane aby spełniały zasady rachunku prawdopodobieństwa i imitowały najdokładniej jak to tylko możliwe środowiska zdarzeń losowych. Jest to prawda przede wszystkim z matematycznego punktu widzenia, jednak w dziele artystycznym istotny jest koncept i sfera ideologiczna i, w moim mniemaniu, istnieje olbrzymia różnica w operowaniu w dziele szumem, który ilustruje działanie algorytmu matematycznego i zegara komputera, a obserwacją szumu w postaci graficznego obrazu ilustrującego chaotyczny ruch elektronów.

Kontemplując wiele przykładów obu rodzajów szumu w postaci graficznych wariacji moich własnych prac, w których wypełniam projekcje szumem w odcieniach szarości, dostrzegam wyraźną różnicę w ich charakterystyce. I może to tylko mój punkt widzenia, ale ten szum, którego początek bierze się z ruchu elektronów, jest nieporównywalnie piękniejszy i zawiera w sobie swoistą, nieopisaną organiczność formy, choć, zdaję sobie sprawę z tego, że dla przeciętnego obserwującego nie musi to mieć znaczenia.

ALGORYTM

Algorytmy pojawiają się w naszym życiu na setki sposobów i często nawet nie uświadamiamy sobie tego, że z nich korzystamy, np. gotując zupę wg. przepisu wykonujemy konkretne instrukcje w celu osiągnięcia określonego efektu. Mamy z nimi do czynienia grając w gry, przeglądając strony internetowe, robiąc zakupy w sieci – za każdym razem, kiedy wchodzimy w obszary związane z technologiami komputerowymi czy mobilnymi. Za umożliwienie nam korzystania z powyższych obszarów odpowiadają specjaliści i hobbyści z dziedzin informatycznych, którzy piszą algorytmy w postaci kodu komputerowego. To wszystko przykłady wykorzystania algorytmów w celu stricto użytkowym, ale są także tacy twórcy algorytmów, dla których one same w sobie stają się celem i obiektem piękna, a ich działanie, wykonanie czy rezultat schodzą na dalszy plan. Taka zależność zachodzi w sztuce, choć także i tu korzystanie z algorytmów odbywa się zarówno świadomie, jak i nieświadomie.

Są sytuacje, w których algorytm pełni rolę wyznaczoną mu przez matematyczne definicje, ale są i takie, w których jego zapis nie będzie prosty do zrozumienia, a sam proces jego działania i wykonania instrukcji nie będzie miał jasno określonego zakończenia. Definicje algorytmu, jakie możemy znaleźć w podręcznikach matematycznych czy informatycznych, wydają się ograniczać to zagadnienie do roli, jaką w tych dyscyplinach odgrywa. Na ogół wymaga się od algorytmu aby prowadził do wykonania zadania w najprostszy możliwy sposób: by był zwięzły i miał określoną liczbę efektów działania i, przede wszystkim, by w określonym momencie kończył swoje działanie.

W obliczu braku pełniejszej definicji algorytmu, obejmującej wszystkie z potencjalnych jego zastosowań, postaram się sformułować własną, o możliwie otwartym charakterze. Najistotniejsze będzie tu zastosowanie tej definicji w dziedzinach sztuki oraz wszelkich działaniach hybrydowych. Moim celem jest dokonanie podziału zastosowań algorytmu w aktywnościach artystycznych, po to, by wyraźnie pokazać jak działanie algorytmu przejawia się moich własnych pracach, szczególnie tych, o których traktuje niniejszy opis.

Definicja i jej objaśnienie

Algorytm – zbiór instrukcji do wykonywania zadania o dowolnej ale logicznej formie przekazu i wcześniej ustalonej, niezmiennej składni.

Tak skonstruowana definicja pozostawia olbrzymią dowolność zakwalifikowania działań w jej obrębie. Mimo zwięzłej formy nadal możemy wyłonić elementy niezbędne do zaistnienia algorytmu. Po pierwsze algorytm musi być zbiorem instrukcji, jednak może to być zbiór o dowolnej liczbie elementów, w tym także zbiór jednoelementowy – posiadający jedną instrukcję.

Po drugie algorytm musi w jakiś sposób zostać przekazany obiektowi wykonującemu, jednak nie określam tu formy przekazu, pozostawiając to zagadnienie otwartym. Oczywiście, w podstawowym rozumieniu, algorytm powinien posiadać zrozumiałą formę zapisu, jednak dopuszczam tu różne formy emisji poza zapisem: werbalną – przekazywaną bezpośrednio lub np. poprzez nagranie, a także dowolny system kodu migowego czy sygnalizacyjnego.

Po trzecie wymieniam niezmiennosc formy przekazu, co uważam za niezwykle istotne. Możliwe, że jest to dość intuicyjne jednak muszę wykluczyć sytuacje, w których instrukcje przekazywane będą w trakcie wykonywania algorytmu w sposób zmienny: forma przekazu algorytmu powinna zostać ustalona przed jego zapisem czy przekazem i musi być zrozumiała dla obiektu wykonującego. Programista przed zapisaniem algorytmu w formie kodu komputerowego musi poznać określony język programowania i rządzące nim zasady po to, by następnie „zrozumiała” je maszyna wykonująca. Podobna zasada reguluje działanie kucharza, chcącego zrealizować przepis na konkretną potrawę.

Celowo nie stawiam wobec algorytmu wymagań skończoności wyników działania, a powoduje to świadomość sytuacji, w których algorytm zawiera instrukcje powrotu do poprzednich instrukcji, tworząc pewnego rodzaju zapętlenie mogące trwać w nieskończoność. Jeśli weźmiemy jako przykład instrukcje wykonania obrazu drogą losową z rozdziału Losowość zobaczymy, że tak skonstruowany algorytm – nie ograniczony rozmiarem płótna i ilością pól – może być teoretycznie wykonywany bezustannie, nie prowadząc do żadnej skończonej liczby rezultatów.

W celu dalszych dywagacji przedstawię prosty podział dotyczący formy występowania algorytmów w dziele artystycznym z uwagi na jego zastosowanie. Podobny podział nie wyczerpuje zagadnienia zastosowania algorytmu w dziele artystycznym, gdyż często mamy do czynienia z sytuacjami, w których działanie artystyczne wykorzystywać będzie algorytm w każdej z przedstawionych ról lub dowolnej ich kombinacji. Przy dalszych rozważaniach istotne będzie jednak to, że poniższy podział nie ma za zadanie opisanie czy też stworzenia na nowo istniejącego już pojęcia “sztuki algorytmu” i klasyfikacji istniejących już dzieł w jej obrębie. Moim założeniem jest jedynie wyłonienie i sklasyfikowanie zastosowań algorytmu w działaniach artystycznych w pojęciu ogólnym.

Algorytm jako narzędzie

Użycie algorytmu jest często nieuświadomione i nawet wiedząc o istnieniu terminu algorytm, czy jego wszechobecnym zastosowaniu w technikach informatycznych i komputerowych, używamy oprogramowania komputerowego nie zastanawiając się nad tym jak ono działa i jakie mechanizmy nim sterują. Wymagamy od technologii aby wykonała określone zadanie, a zasadami jej działania interesują się nieliczni z nas. Korzystamy z gotowych rozwiązań – w tym przypadku algorytmów – stworzonych przez bliżej nieznane osoby trzecie w charakterze narzędzia i nie zadajemy sobie pytania, z czym mamy do czynienia. Wszelkie programy komputerowe do edycji grafiki komputerowej są w istocie bardzo skomplikowanymi algorytmami, nie wspominając nawet o technologiach internetowych. Do omawianej tu kategorii chciałbym zakwalifikować wszelkie działania wykorzystujące oprogramowanie komputerowe w celu uzyskania określonego efektu artystycznego graficznego czy interaktywnego. Ważne jest tutaj oddzielenie sfery dzieła, jego postaci graficznej, formalnej czy konceptualnej, od działania algorytmu. Tu algorytm nie jest ani dziełem, ani zasadą jego działania. Jest narzędziem w ręku artysty, jak dłuto w ręku rzeźbiarza, które samo w sobie wykonane zostało przez rzemieślnika i nie niesie ze sobą żadnej strefy konceptualnej wzbogacającej dzieło.

Algorytm jako instrukcja

W tym przypadku algorytm wymyka się z obszaru definicji informatycznych i pełni rolę instrukcji dla artysty lub też instrukcji artysty dla odbiorcy. Jest zapisem conceptualnej sfery dzieła, która może przeobrazić się we wszelkiego rodzaju obiekty czy też działania artystyczne. Tu znów można posłużyć się przykładem z rozdziału Losowość – obrazu tworzono za pomocą losowych rzutów kością lub monetą wg określonych zasad czyli algorytmu. Jest przecież znanym faktem to, że Ryszard Winiarski tworząc swoje dzieła na bazie mechanizmów losowych działał wg. określonych instrukcji, które, spisane na kartce, umieszczał po drugiej stronie obrazu wykonanego wg. danego algorytmu. W takich przypadkach algorytm może zostać wykonany wielokrotnie dając czasem podobne, a czasem całkiem różne rezultaty graficzne, jednak to właśnie algorytm odgrywa decydującą rolę w kreowaniu dzieła. Jest swojego rodzaju matrycą, sztancą, czy też wzorem określającym granice wariacji jego formy. Z drugiej strony w tej kategorii występować będą czysto conceptualne działania pozbawione efektów w świecie rzeczywistym, może poza formą zapisu samego algorytmu bądź takimi przykładami, które skutkować będą działaniami efemerycznymi i krótkotrwałymi. Do takich zjawisk z dziedziny sztuki można zaliczyć np. flash mob, którego początkiem zawsze są spisane przez autora instrukcje dla uczestników. Znamy też przykłady dzieł conceptualnych mających postać algorytmu – instrukcji skierowanych do odbiorcy, nie zawsze wymagających ich wykonania, a jednak niosących istotne założenia ideowe. Podobnie rzecz się ma z sięgającym początków conceptualizmu pojęciem event score - scenariuszy wydarzeń o charakterze performatywnym w formie prostych instrukcji pozbawionych innych charakterystycznych dla scenariusza elementów jak np. dialogi.

“Compositions 1960” La Monte Young’a to zestaw instrukcji o dostrzegalnej formule algorytmu, często niewykonalnych dla odbiorcy (co nie powstrzymało Nam Jun Paika przed interpretacją jednej z nich “draw a straight line and follow it” z wykorzystaniem nocnika z atramentem i krawata na własnej szyi). Kolejnym przykładem tego rodzaju dzieła jest książka “Grapefruit” składająca się z prostych instrukcji, z której każda może (ale nie musi) zostać wykonana – genialne działanie conceptualne autorstwa Yoko Ono.

TUNAFISH SANDWICH PIECE

Imagine one thousand suns in the
sky at the same time.

Let them shine for one hour.

Then, let them gradually melt
into the sky.

Make one tunafish sandwich and eat.

– 1964 Spring

Podsumowując, w tej kategorii klasyfikuję prace, w których algorytm jest nośnikiem sfery conceptualnej i ideowej dzieła, a efekt jego działania może – choć nie musi – zaistnieć w sferze rzeczywistej. Algorytm pojawia się tutaj w formie wyraźnych instrukcji, a charakter jego zapisu nie wpływa na jego odbiór.

Algorytm jako dzieło

W tym przypadku obiektem artystycznym jest sam algorytm: nie jego instrukcje, nie sfera konceptualna jaką niesie, nawet nie efekt jego działania, a właśnie postać algorytmu sama w sobie. Skutkiem działania takiego algorytmu może być wyświetlenie na ekranie komputera napisu o dowolnej treści, dokonanie prostego obliczenia lub całkiem skomplikowanych działań graficznych i to właśnie forma jego zapisu eksponowana będzie jako dzieło. Jako przykład można podać działania artystów którzy wykorzystując kod komputerowy, tworzą instalacje interaktywne i animacje proceduralne lub dowolne formy graficzne z wykorzystaniem języków programowania, a jednocześnie obok tych prac, jako istotny ich element, eksponują sam kod w formie wydruku czy plakatu. Kod komputerowy ma swoją charakterystykę graficzną i typograficzną, powtarzalną strukturę, którą można uznać za niezwykle estetyczną. W dodatku kod jako układ tekstu może być niemal dowolnie przekształcany, jako że tzw. "białe znaki" nie mają w nim żadnego znaczenia. Możemy taki kod pozbawić zupełnie spacji, tabulatorów i znaków końca linii, a wciąż będzie wykonywalny jako instrukcja dla komputera. Ta dowolność w formatowaniu kodu wprowadza w dziedzinę formalnego algorytmu komputerowego miejsce na zmiany autorskie w wyglądzie samego kodu – doświadczeni programiści miewają swoje przyzwyczajenia w formatowaniu kodu, które potrafią stać się ich znakiem rozpoznawczym.

Skoro taka ingerencja jest możliwa oczywistym jest, że artyści posługujący się językami programowania jako medium chętnie docenią graficzną reprezentację algorytmu, który uosabia ich myśl od strony ideowej. Zapis taki nie jest instrukcjami dla samego artysty ani instrukcjami dla widowni. W tym wypadku kod komputerowy wykonuje lub wykonał swoją pracę jako narzędzie lub mechanizm sterujący dziełem, a jego wydruk staje się artefaktem artystycznym zawierającym – ale nie eksponującym – sfery konceptualnej dzieła. Dla znakomitej większości odbiorców nie znających zagadnień informatyki ukazany w tej formie algorytm nie ma innej wartości poza czysto estetyczną i na pewno zagadkową i intrygującą formą.

Podobne działania wykorzystywał Joseph-Nechvatal w swoim projekcie artystycznym "Computer Virus Project". Algorytm komputerowy wystąpił tu w roli wirusa degenerującego postać graficzną cyfrowych reprodukcji wcześniejszych prac artysty, jako symulacja komputerowa. Jednocześnie wraz z animacją działania wirusa na wystawie eksponowany był algorytm sterujący wirusem w postaci zapisu kodu komputerowego, na ogromnym, ponad czterometrowej szerokości płótnie w orientacji poziomej (czytanej od góry do dołu, a nie zwyczajowo od lewej do prawej) co dodatkowo odbierało mu walory czytelności i umiejscawiało go właśnie w roli artefaktu graficzno-typograficznego.

Dla wielu artystów posługujących się algorytmem komputerowym w postaci języków programowania kod sam w sobie staje się obiektem podziwu i dla wielu dzieło istnieje w tym właśnie kodzie i to on jest jego istotą a nie efekt końcowy jego działania. Jest też dziedzina informatyki, w której algorytm komputerowy wyzwolił się z ograniczeń definicji wymagającej od niego prostoty, a języki programowania powstają tam nie po to aby służyć użytkownikowi – będąc możliwie najprostszymi w obsłudze – a wręcz przeciwnie! Autorzy chcą, aby stworzone przez nich języki programowania były wyjątkowe, trudne w zrozumieniu, prześmiewcze i, przede wszystkim, inne. Mam tu na myśli najbardziej niszową dziedzinę spośród niszowych dziedzin

świata programistycznego, czyli ezoteryczne języki programowania. W moim mniemaniu sfera ta, istniejąc na pograniczu sztuki i informatyki, nie doczekała się nadal należnego opracowania od strony artystycznej wartości konceptualnej.

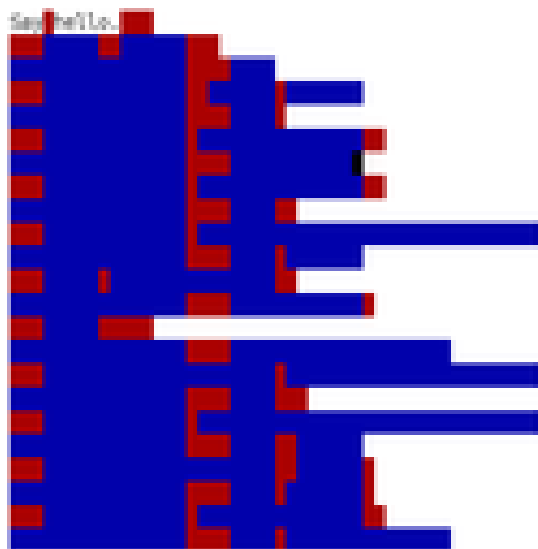
Kod komputerowy służy do wydawania komend komputerowi przez programistę. Kod komputerowy posiada formę lingwistyczną tylko w jednym celu: aby być zrozumiałym dla człowieka. Kod komputerowy napisany przez człowieka jest tłumaczony przez oprogramowanie komputerowe zwane "kompilatorem" na kod maszynowy zrozumiały dla komputera i o ile zachowane będą zasady logiki w zapisie od strony człowieka, kod może przybrać całkiem dowolną tekstową, cyfrową lub graficzną formę, zakładając, że może zostać przetłumaczony przez kompilator na język maszynowy.

Przedstawię poniżej przykłady algorytmu komputerowego w postaci różnych ezoterycznych języków programowania. Efekt działania tych algorytmów będzie bez znaczenia (z reguły jest to kod generujący na ekranie komputera zwyczajowy w informatyce napis "Hello World"). Postać zapisu algorytmu jest tutaj w moim mniemaniu graficznym jak i konceptualnym dziełem sztuki.

Język *Brainfuck*

```
+++++++>++++>[>+>+>+>+>+>+<<<<-]  
>+>+>->>+<[<-]>>.>---.+++++++..+++>>.  
<-.<..+++..-----..-----..>>+>+>.
```

Język *Whitespace*



Język *Piet*



Każdy z powyższych przykładów to pełnoprawny język programowania, w którym kod przybiera formę symboli, znaków bądź wartości kolorystycznych, które w świecie programistycznym są po prostu wartościami liczbowymi i mogą przenosić dowolne informacje.

WZÓR

Wzór może się wydawać pojęciem bardzo zbliżonym do pojęcia algorytmu, jednak błędem byłoby ujednocianie ich znaczeń ponieważ, podczas gdy algorytm jest zbiorem instrukcji, wzór, zgodnie z założeniami matematyki, fizyki czy chemii, jest opisem zjawiska wyrażonym symbolami. Wzór objaśnia dane zjawisko niezmiennie, np. wzór na pole kwadratu określa każdy istniejący kwadrat w dowolnej postaci, a wzory fizyczne opisują niezmiennie zjawiska i właściwości świata materialnego.

Wzór może stać się początkiem stwarzania algorytmu potwierdzającym jego działanie, ale sam w sobie nie jest instrukcją, a jedynie symboliczną reprezentacją działania. Algorytm będzie jego rozwiązaniem.

Jako przykład podajmy jeden z najbardziej znanych wzorów na kolejne wyrazy w ciągu Fibonacciego, którego zasadą jest to, że każdy kolejny wyraz ciągu jest sumą dwóch jego poprzednich wyrazów. Graficzną reprezentacją tego ciągu jest charakterystyczna spirala wyznaczająca złoty podział płaszczyzny.

Pomijając liczby 0 i 1, dla których wartości tego ciągu wynoszą kolejno 0 i 1, każdy kolejny wyraz ciągu możemy wyliczyć ze wzoru:

$$F_n = F_{n-1} + F_{n-2}$$

Podstawiając w miejsce "n" następujące po sobie cyfry od 2 do dowolnej liczby, uzyskamy kolejne wyrazy ciągu, czyli niezmienny zbiór liczb opisany tym wzorem :

0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181...

Przyjrzyjmy się teraz algorytmowi zawierającemu konkretne kroki do uzyskania tego samego wyniku:

- Weź ciąg cyfr 0,1
- Dodaj dwie ostatnie cyfry uzyskanego ciągu do siebie
- Uzyskany wynik zapisz jako ostatni wyraz ciągu
- Powrót do punktu 2

Podane przykłady doskonale ilustrują różnice między wzorem, a algorytmem i charakteryzują algorytm jako pewnego rodzaju tłumaczenie wzoru na zestaw instrukcji. Instrukcje te w prosty sposób przełożyć można na kod komputerowy. Poniżej przedstawię przykład programu w języku Processing wypisującego w konsoli komputera kolejne 10 wyrazów ciągu Fibonacciego:

```
int F0 = 0;
int F1 = 1;
int Fn;
int liczbaWyrazow = 10;

//poniżej pętla wykonywana tyle razy ile wynosi zmienna "liczbaWyrazow"
for( int i = 1; i <= liczbaWyrazow; i++ )
{
println( F1 ); //komenda drukująca linijkę tekstu z wynikiem eng. "print line"
Fn = F0 + F1;
F0 = F1;
F1 = Fn;
}
```

Według podobnego schematu we własnych pracach używam i przekształcam istniejące już wzory matematyczne za pomocą algorytmów komputerowych, dla własnych potrzeb, generując określone wzory geometryczne: koła, linie, krzywe Beziera o charakterze proceduralnym. Tu rozwinięcia i wytłumaczenia będzie wymagało zastosowanie terminu procedury i proceduralności, co stanie się tematem kolejnego rozdziału.

PROCEDURA

Innym znaczeniowo podobnym terminem do algorytmu jest procedura, rozumiana jako określony sposób postępowania. Procedura w opozycji do algorytmu opisuje raczej zachowanie w określonych sytuacjach, choć, oczywiście, można ją zapisać w postaci algorytmu działania. Procedura może więc być algorytmem lub jego częścią wywoływaną w określonym przypadku. Od niedawna w świecie grafiki komputerowej zaczął się formować i rozpowszechniać termin proceduralności: sformułowania grafika i animacja proceduralna zaczęły powoli wypierać takie terminy jak animacja czasu rzeczywistego czy animacja generatywna. Istotne jest, że wszystkie z wymienionych terminów mają pewną wspólną ideę, którą chciałbym przybliżyć w tym rozdziale.

Wprowadzenie możliwości programistycznych w dziedzinę grafiki komputerowej wyświetlanej za pośrednictwem ekranu dało niespotykane dotąd możliwości informatykom, hobbystom, twórcom gier komputerowych, a także artystom, szczególnie w dziedzinie animacji. Tradycyjna animacja zawsze wymagała olbrzymiego nakładu pracy. Raz ustalona sekwencja animacji była niezmienna w swojej formie i nie pozostawiała miejsca na improwizację. Jej twórca, podobnie jak osoba, która raz już zobaczyła taką animację, doskonale wiedzieli jak wyglądać będzie ona następnym razem.

Wprowadzenie do dziedziny animacji możliwości programistycznych pozwoliło wykorzystać algorytmy pseudolosowe w opisie właściwości animowanych obiektów. Animowane obiekty nie musiały już poruszać się po raz wytyczonych trasach, zyskując niemal autonomiczne życie. Często sam autor nie mógł przewidzieć ich ruchu (wystarczyło, że pisząc algorytm sterujący pozwolił na taką dowolność). Każda klatka zaprogramowanej animacji może być niemal całkiem nieprzewidywalna i niepowtarzalna, "losowa", i to jest główna cecha, która odróżnia ją od animacji tradycyjnej i jakiegokolwiek montażu wideo. Dla oddzielenia od siebie tych dwóch zjawisk potrzebny był nowy termin. Jednym z pierwszych było "realtime animation" w języku polskim – animacja czasu rzeczywistego. Miało to oznaczać, że w przeciwieństwie do tradycyjnych technik, decyzja o tym, co pojawi się w następnej klatce animacji dokonuje się w momencie jej obserwacji.

Zagadnienie grafiki czy animacji proceduralnej opisuje jeszcze bardziej złożony proces generacji obrazu w czasie rzeczywistym. Posłużę się tutaj przykładem ze świata o charakterze bardziej komercyjnym, niż sztuka, chociaż sam pomysł uznaję za doskonały przykład artystycznej myśli konceptualnej.

W 2016 roku na rynku gier ukazała się nowa produkcja pod nazwą No Man's Sky angielskiego studia Hello Games, rewolucjonizując sposób kreacji świata, w którym porusza się gracz. Zwyczajowo wysokobudżetowe gry produkowane są przez studia zatrudniające dziesiątki, a nawet setki grafików komputerowych modelujących każdy, najmniejszy element świata gry. No man's sky jest grą 3D, science fiction, w której gracz porusza się w bezkresnym świecie kosmosu od-

wiedząc planety, kolekcjonując niezbędne mu materiały do budowy bazy i ulepszania ekwipunku postaci. Obliczono, że chcąc odwiedzić każdą z dostępnych planet przez chociażby 1 sekundę, gracz musiałby poświęcić około 5 bilionów lat czasu realnego! A każda z tych planet jest unikatowa, posiada niepowtarzalną florę, faunę i geologię. Stworzenie podobnej zawartości – nawet z armią grafików – byłoby niewykonalne w ciągu 100 lat.

Jak więc trzyosobowe studio dokonało tego cudu? Otóż dzięki algorytmom, wzorom i procedurom! Powstał genialny algorytm stworzenia wszechświata wirtualnego, w którym na podstawie prostych liczbowych danych wejściowych (ziaren) wokół gracza generowane jest otoczenie – przestrzeń międzyplanetarna, stacje kosmiczne, asteroidy i w końcu same planety z całą ich zawartością. Jako ziarna użyto liczb współrzędnych X, Y, Z (położenia postaci gracza w rzeczywistości trójwymiarowej gry), co w dodatku pozwala zachować spójność świata, ponieważ każdy z graczy znajdujący się w obrębie tych samych koordynatów zobaczy dokładnie tę samą zawartość! To jest powód, dla którego nie można tu mówić o generowaniu losowym, ponieważ położenie każdego elementu współtworzącego wszechświat gry jest z góry ustalone przez algorytm.

Mamy tu do czynienia jednak z olbrzymią skalą nieprzewidywalności efektów. Sami twórcy nie widzieli nawet promila wszechświata, który stworzyli, a algorytm generujący jest na tyle skomplikowany, że jego autorzy nie są w stanie przewidzieć, co może znaleźć się parę kroków od centrum tego wszechświata.

Proceduralność jest więc terminem doskonale opisującym zjawiska z dziedziny grafiki komputerowej obejmujące wszelkie formy generacji pseudolosowej czasu rzeczywistego i zawierające trzy podstawowe składniki:

- Ziarno – dane wejściowe (na ogół alfanumeryczne)
- Algorytm sterujący – zestaw instrukcji przetwarzających dane wejściowe w efekt końcowy (przeważnie w postaci kodu komputerowego)
- Efekt końcowy (najczęściej w postaci graficznej animacji komputerowej)

Dodatkowo można uściślić, że w przypadku proceduralności raz ustalony algorytm sterujący reaguje na dane wejściowe w ten sam sposób. Idąc tym tropem, gdybyśmy wrócili do pierwszego rozdziału i spojrzeli na przykład pierwszego algorytmu generującego liczby pseudolosowe na podstawie dowolnego ziarna zobaczymy, że ma on charakter proceduralny i generowane liczby możemy również nazwać „proceduralnie generowanymi o określonym stopniu nieprzewidywalności”.

W moim mniemaniu jest to wystarczający powód ku temu, by odejść od jednoznacznego kwalifikowania prac o charakterze proceduralnym w terminologii losowości. Postrzegam losowość jako ideę, do której można dążyć, ale nigdy osiągnąć. Prawdziwa losowość jest dla mnie opisem stopnia nieprzewidywalności, który, gdyby zaistniał w świecie rzeczywistym, stworzyłby realną szansę – przy wielokrotnej próbie – odpalenia silnika spalinowego po wlaniu wody do baku zamiast benzyny.

OPIS PRAC I ZASTOSOWANYCH TECHNOLOGII

Aby móc studiować procesy pseudolosowe o wysokim stopniu nieprzewidywalności na dużej liczbie przypadków kompozycji malarskiej musiałem wyeliminować udział człowieka (artysty) jako możliwej dodatkowej zmiennej w działaniu algorytmu.

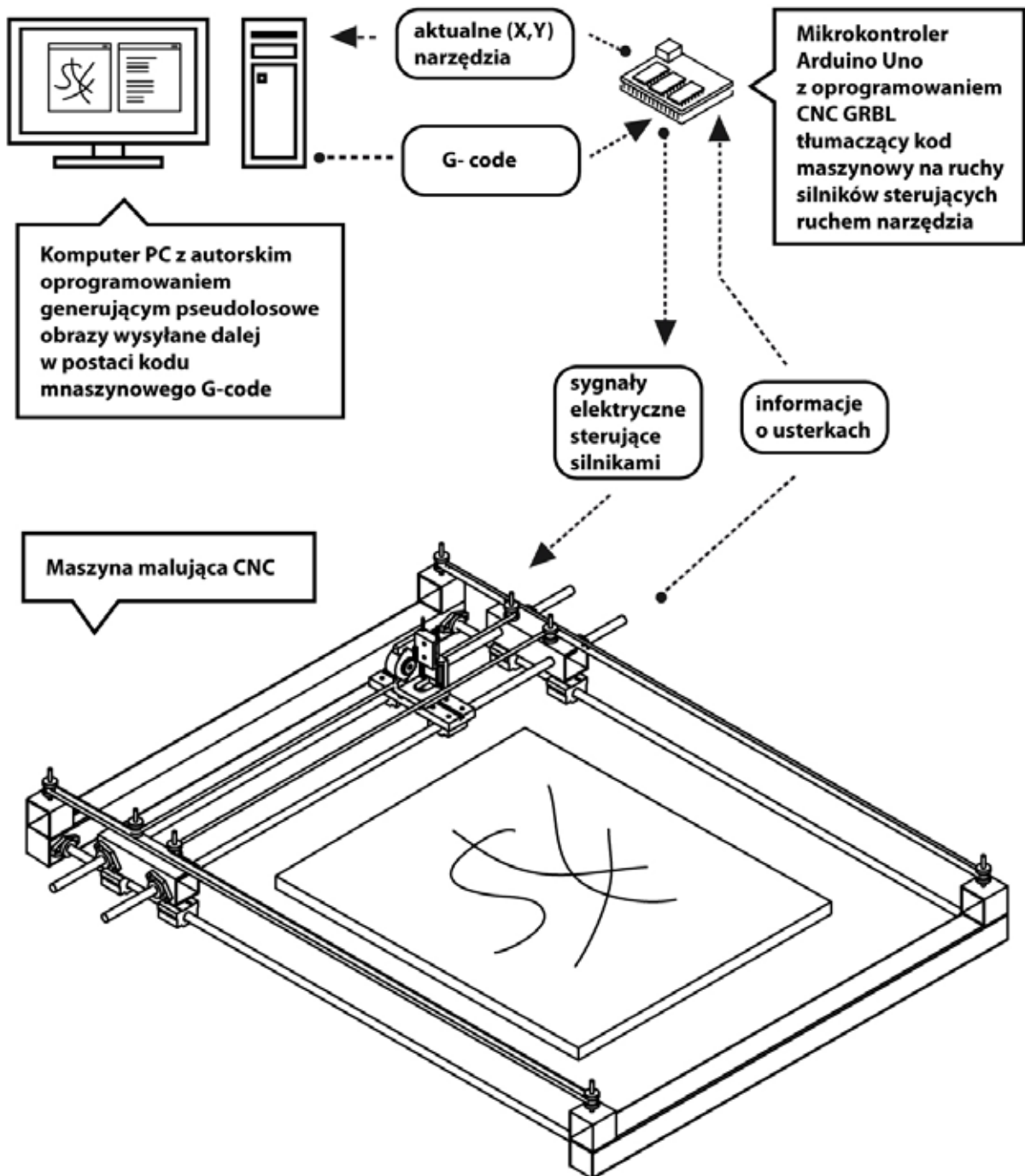
Po rozłożeniu na akcje proste procesu powstawania obrazu w technikach tradycyjnych dostrzegamy że dłoń artysty porusza się w 3 wymiarach obszaru roboczego - osiach X i Y płaszczyzny podobrazia oraz w osi Z - kontrolując nacisk czy pobierając kolor medium. Aby wyeliminować udział artysty w procesie powstawania obrazów musiałem stworzyć mechaniczne środowisko pozwalające narzędziu malarskiemu poruszać się w podobny sposób w osiach X, Y, Z przy możliwie najprostszymi środkami technologicznymi.

Rozwiązaniem które spełnia powyższe wymagania było zastosowanie maszyny CNC - technologii rozpowszechnionej i spopularyzowanej w ostatnich latach na wielką skalę, przez społeczność konstruktorów, hobbystów i jak i specjalistycznych dostawców maszyn do druku 3D i maszyn tnących.

W dużym uproszczeniu konstrukcja CNC opiera się na trzech osiach transmisyjnych (dla każdej z osi X, Y, Z) pozwalających przemieszczenie głowicy z narzędziem w dowolny punkt osi X, Y i ruchu w osi Z, podnoszącym i opuszczającym narzędzie (w moim przypadku medium tradycyjne w postaci pędzla nanoszącego farbę akrylową).

W maszynach CNC ruchem głowicy steruje oprogramowanie zainstalowane na komputerze przesyłające do maszyny kolejne koordynaty X Y Z zaczerpnięte z pliku projektu przygotowanego wcześniej za pomocą dowolnego oprogramowania graficznego. Powiedzmy że chcemy namalować koło, wystarczy wówczas przygotować rysunek np. w programie Illustrator (w skali 1:1) i zaimportować plik do oprogramowania maszyny. Narzędzie maszyny ustawione w pozycji spoczynku w dowolnym narożniku podnosi się w osi Z i nie dotykając płótna wędruje do pozycji startowej, narzędzie opuszcza się. Następnie oprogramowanie wysyła do maszyny kolejne koordynaty XY odpowiadające kolejnym punktom na płaszczyźnie wyznaczającym okrąg. Po wyrysowaniu okręgu narzędzie powraca do pozycji spoczynku lub, jeśli wyznaczono kolejne, zadanie pobiera porcję farby wędrując do zbiornika z farbą i następnie kontynuuje pracę. Oprogramowanie sterujące dla maszyn tego typu jest ogólnie dostępne a nawet darmowe za pośrednictwem licencji otwartych, jednak podobnie do przytoczonego przykładu pozwala przenosić kompozycje z przygotowanych wcześniej plików. W przypadku mojej pracy konieczne było stworzenie oprogramowania podejmującego decyzje o kolejnym kroku maszyny w czasie rzeczywistym.

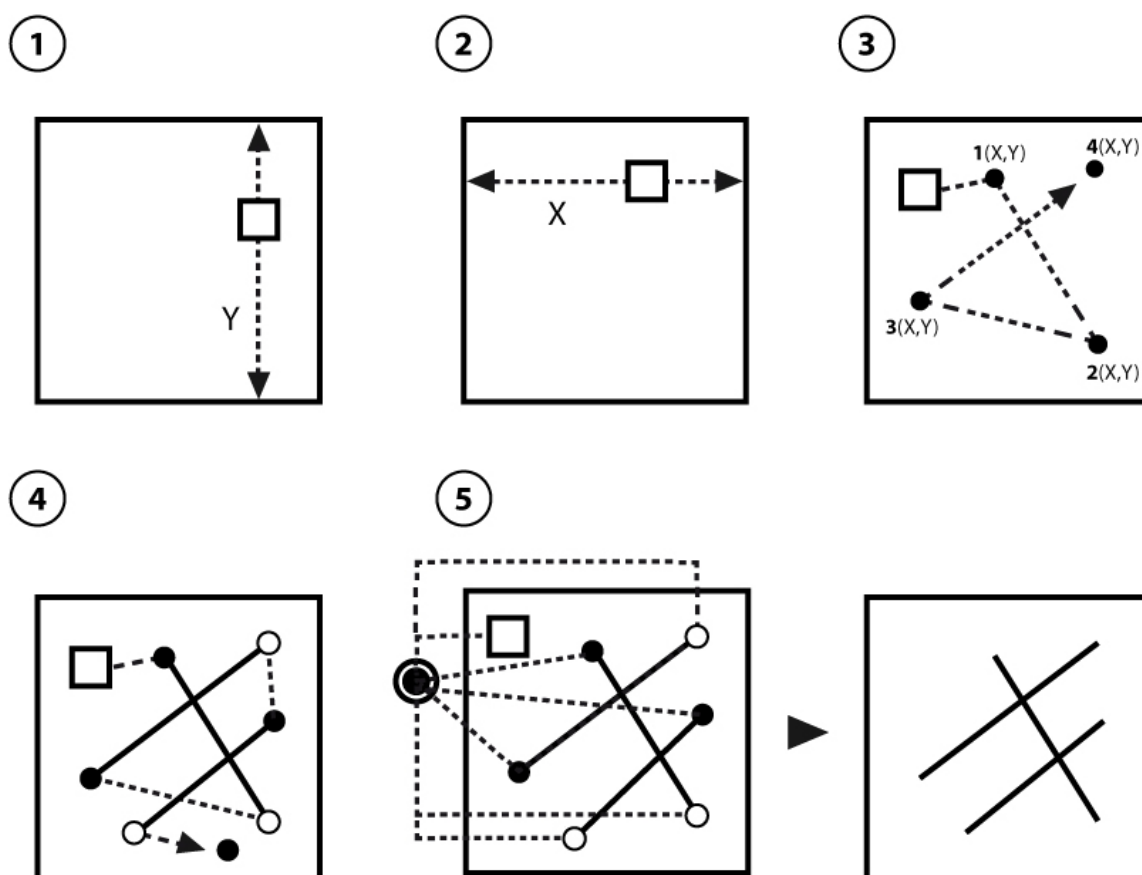
Poniższy schemat przedstawia ideową postać działania wykonanej przeze mnie maszyny w parze z komputerem i zastosowanym oprogramowaniem:



Pierwszą czynnością po stworzeniu fizycznej struktury maszyny, było stworzenie podstawowych możliwości poruszania narzędziem (rysującym lub malującym) w obrębie obszaru roboczego (płótna) przez oprogramowanie podstawowe (nie będące jeszcze algorytmem koncepcyjnym a zestawem maszynowych instrukcji).

Oto podstawowe kroki, które wspomniane oprogramowanie musiało spełnić, przy założeniu że obszar roboczy to płaski układ o osiach X i Y:

1. ruch w osi Y
2. ruch w osi X
3. ruch w obu osiach naraz po kolejno wyznaczonych punktach (X, Y)
4. opuszczanie i podnoszenie narzędzia – ruch w osi Z – rozróżnienie, kiedy narzędzie podróżuje do miejsca początku nowej linii, a kiedy faktycznie upuszcza pędzel na płótno i maluje
5. w przypadku pędzla w roli narzędzia należało przewidzieć konieczność podróży narzędzia do zbiornika z farbą i późniejszej kontynuacji malowanego wzoru.



Po tak przygotowanym podstawowym programie sterującym i sprawdzeniu jego funkcjonalności przystąpiłem do konstrukcji algorytmu mogącego generować kształty i wzory geometryczne przesyłane do maszyny jako kolejne punkty-koordynaty, w układzie współrzędnych Y, X.

Przed zamianą na kod komputerowy stworzyłem algorytm ideowy zakładający wyposażenie zespołu oprogramowanie-maszyna w możliwości generowania podstawowych, jak i skomplikowanych kształtów geometrycznych. Posługując się różnymi wzorami matematycznymi uzyskałem algorytmy generujące kształty geometryczne o zmiennych własnościach, które w dalszym procesie mogły zostać stworzone na podstawie zmiennych pseudolosowych. Po wyposażeniu oprogramowania w umiejętność generowania określonych kształtów i wzorów

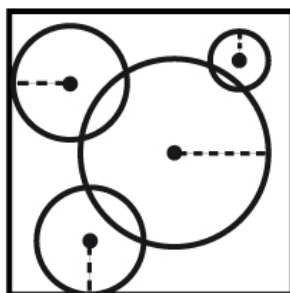
geometrycznych, pozostało stworzenie algorytmu losującego – decydującego, która z dostępnych umiejętności zostanie użyta i jak wiele razy.

W celu osiągnięcia bardziej skomplikowanych wzorów, chociażby organicznych form, musiałem skorzystać z gotowych formuł matematycznych z zakresu geometrii obliczeniowej i przełożyć je do postaci algorytmu komputerowego sterującego maszyną. Po wyposażeniu maszyny w narzędzie umożliwiające podróżowanie z pkt. A do pkt. B można było generować proste geometryczne kształty (1), aby jednak mogły powstać elipsy i koła musiałem skorzystać ze wzoru na położenie obiektu orbitującego (2). Kolejnym istotnym narzędziem były losowo generowane linie Beziera, zastosowanie których pozwoliło na sterowanie krzywizną linii (3). Zastosowałem także wzory na odnajdywanie tzw. otoczki wypukłej, co umożliwiło mi na generowanie bardziej skomplikowanych i dość przypadkowych brył (4), oraz na triangulację Delaunay – wyszukiwanie podziału zestawu punktów do siatki trójkątów (5). Ostatnią wybraną przeze mnie subrutyną było stworzenie algorytmu mogącego wypełniać podany kształt równoległymi liniami tworzącymi w efekcie w miarę dokładne plamy kolorystyczne (6).

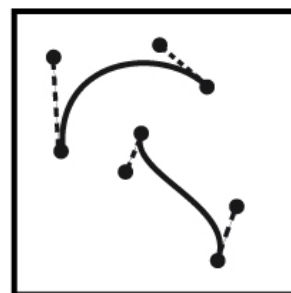
1



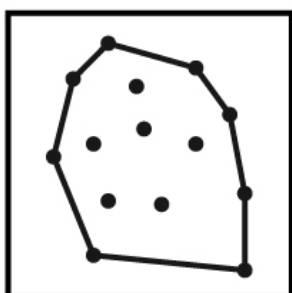
2



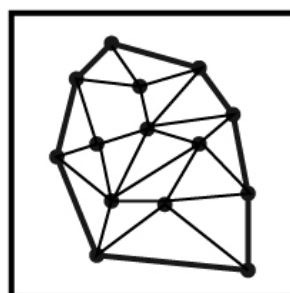
3



4



5



6



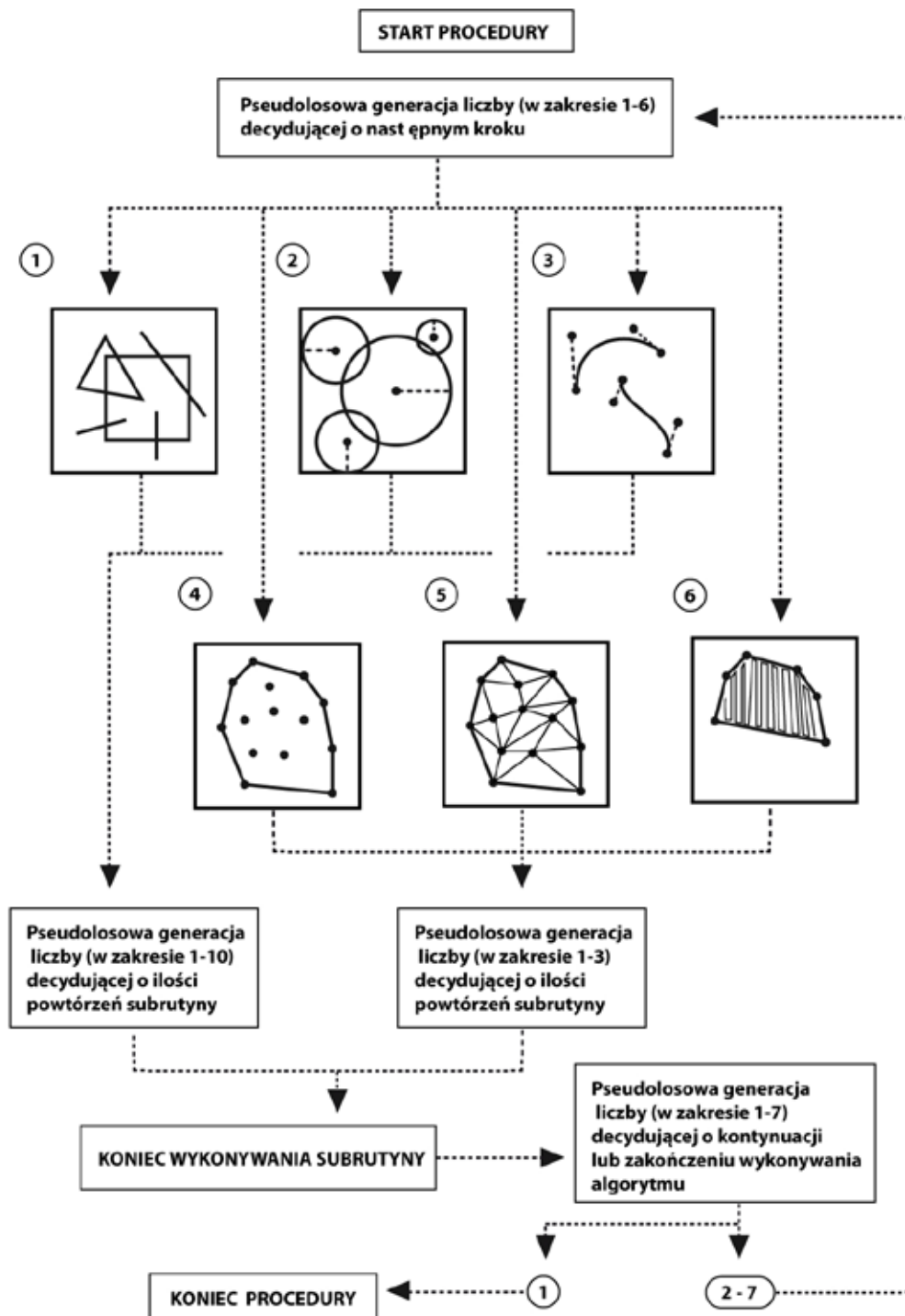
Proces wyposażania oprogramowania w odpowiednie algorytmy rysujące przypominał proces uczenia dziecka rysunku kolejnych kształtów geometrycznych, teraz należało dać temu dziecku załóżek świadomości a przynajmniej zarys procesu decyzyjnego - co będzie teraz malowane - w oparciu o pseudolosowość. Duża liczba subrutyn i ogromna liczba możliwości ich kombinacji, w dodatku duża nieprzewidywalność narzędzia malarskiego w postaci pędzla i farby stworzyły wysoki stopień nieprzewidywalności w kompozycji obrazu.

Z tego powodu postanowiłem chwilowo zrezygnować z dodatkowej komplikacji procesu generacji danych losowych i użycia generatora szumu na rzecz zwykłych liczb pseudolosowych generowanych przez komputer.

W trakcie eksperymentowania z różnymi formami algorytmu przez różne jego etapy wyłoniłem dwa tryby pracy z kompozycją i decyzjami kompozycyjnymi algorytmu:

- Algorytmy pseudolosowe
- Algorytmy sekwencyjne

Poniższy diagram ilustruje procesy decyzyjne ostatecznej formy pseudolosowego algorytmu sterującego moją pracą.



Poniżej wyjaśnię działanie algorytmu sekwencyjnego, który stał się ważną częścią mojej pracy choć pojawił się poniekąd "przypadkowo" a może raczej w wyniku eksperymentu i nie jest on najważniejszą częścią mojej pracy a jednak widzę w tym zagadnieniu olbrzymi potencjał.

Pracując nad losową deformacją kształtu geometrycznego jak na przykład koło przez algorytm szumu (w szczególności szumu Perlina autorstwa Kena Perlina) odkryłem możliwości sekwencyjnego przekształcenia formy geometrycznej. Algorytm ten działa w podobny sposób do algorytmu pseudolosowego opisanego przeze mnie w rozdziale Losowość. Korzystając z podanego przez użytkownika ziarna generuje on pseudolosowe dane wyjściowe i oczywiście podając to samo ziarno uzyskujemy ten sam efekt. Przepuszczenie punktów (ich współrzędnych) składowych okręgu lub wielokąta przez podobny algorytm dla określonego ziarna, skutkuje określoną deformacją tego kształtu. W ten sposób dla np. liczby 1 uzyskujemy powtarzalną deformację okręgu i zawsze przy tym samym ziarnie będzie wyglądała tak samo, ale już dla liczby 2 dwa będzie ona wyglądała całkiem inaczej. Powstała w mojej głowie idea odnalezienia takich przekształceń dla możliwie największej ilości liczb i postawienia tezy, że przy nieskończonej liczbie przekształceń (spowodowanej nieskończonością liczb ziarna) jedna liczba może opisywać dowolny kształt, i że może istnieć liczba opisująca obrys profilu mojej twarzy, granice dowolnego kraju czy najnowszego modelu samochodu dowolne firmy. Widzę w podobnym założeniu sekwencyjność i seryjność obrazów Opałki czy kombinatoryczne wszechświaty możliwości graficznych Every Icon Johna F. Simona Jr.

Działanie wytłumaczonych powyżej algorytmów, poza zmianami narzędzia i decyzjami dotyczącymi gamy kolorystycznej decyduje w pełni o postaci powstającego obrazu. Zrozumienie mechanizmów działania skonstruowanego przeze mnie systemu malarskiego pozwala zrozumieć samą ideę tegoż eksperymentu jako dzieła malarskiego i artystycznego.

Wybrany zestaw obrazów to reprezentacja części dzieł powstałych w trakcie procesu badawczego działania maszyny. Znajdujemy tu zarówno testy na wyizolowanych subrutynach – linie Beziera, linie proste, triangulacja Delaunay itd., jak i wyniki działania pełnego algorytmu łączącego te procesy, tworzące coraz bardziej nieprzewidywalne wartości malarskie. Ponadto, testowanie efektu graficznego zakładało też zmianę samego narzędzia z tradycyjnego medium malarskiego w postaci pędzla i farby akrylowej, do różnego rodzaju pisaków i flamastrów, jak i eksperymentów z kolorystyką jednego i drugiego medium.

Niezmiennie podkładem i tłem dla kompozycji zostaje w każdej z prac białe płótno 50x50 cm pełniące rolę swojego rodzaju szalki Petriego co pozostaje w zgodzie z laboratoryjnymi założeniami eksperymentu.

KONKLUZJA

Za pośrednictwem algorytmów, wzorów, procedur i całego arsenału rozwiązań technologicznych, poszukuję graficznych i malarskich form reprezentacji zdarzeń o wysokim stopniu nieprzewidywalności. Każda nowa forma ekspresji artystycznej jest dla mnie eksperymentem, badaniem, eksplorowaniem nowej estetyki. Poszukuję nowych wzorów i algorytmów i sposobów generowania dzieł w nieustannej próbie odnalezienia czegoś, czego nawet we własnej wyobraźni nie mógłbym odnaleźć. Technologie komputerowe są dla mnie rozszerzeniem gamy mediów tradycyjnych. Malując, rzeźbiąc, czy posługując się technikami tradycyjnymi nie potrafiłem uciec od swojego rodzaju oprogramowania będącego istotną częścią mojej osobowości. Doświadczenie życia, ograniczenie fizycznej jego postaci, wpojony program nauczania, biologiczne i psychiczne potrzeby w jakiejś formie sterowały każdym ruchem mojej ręki: tworząc kolejne dzieło, nie odnajdowałem nic nowego. Nawet ucieczka od estetyki obiektu do form konceptualnych była efektem tego samego oprogramowania. Nie mogąc uwolnić się od siebie samego zacząłem szukać sposobu na oddzielenie siebie od większości procesów decyzyjnych w powstawaniu dzieła. Tworząc kreowałem swoiste laboratoria, w których mogłem wyznaczać graficzne granice i obserwować bogactwo zdarzeń w ich ramach. Tak zaprogramowany algorytm nigdy nie będzie odnosił się do form realistycznych.

Z fascynacją obserwuję jak algorytm generujący wzory, które uznaję za estetyczne, jest w stanie stwarzać także takie, których, w najlepszym przypadku, nie uznałbym za dzieło skończone. To pozwala mi wyrwać się z wpojonego świata estetyki i przełamać jego ograniczenia, dostrzec piękno w bogactwie iteracji formy.

Powstałe prace i algorytmy, to jedynie załączek możliwości wykonanych i użytych przeze mnie rozwiązań technologicznych. W trakcie prac nad ich obecną formą powstawały w mojej głowie dziesiątki nowych pomysłów, rozwiązań i idei, których nie zdążyłbym wypróbować i zastosować. Kończąc ten etap prac mam pewność, że nareszcie mam do czynienia z medium, którego możliwości mogą poszerzać moje spektrum kreacji do tego stopnia, w którym w nie będę mógł przewidzieć całkiem efektu końcowego. W końcu!

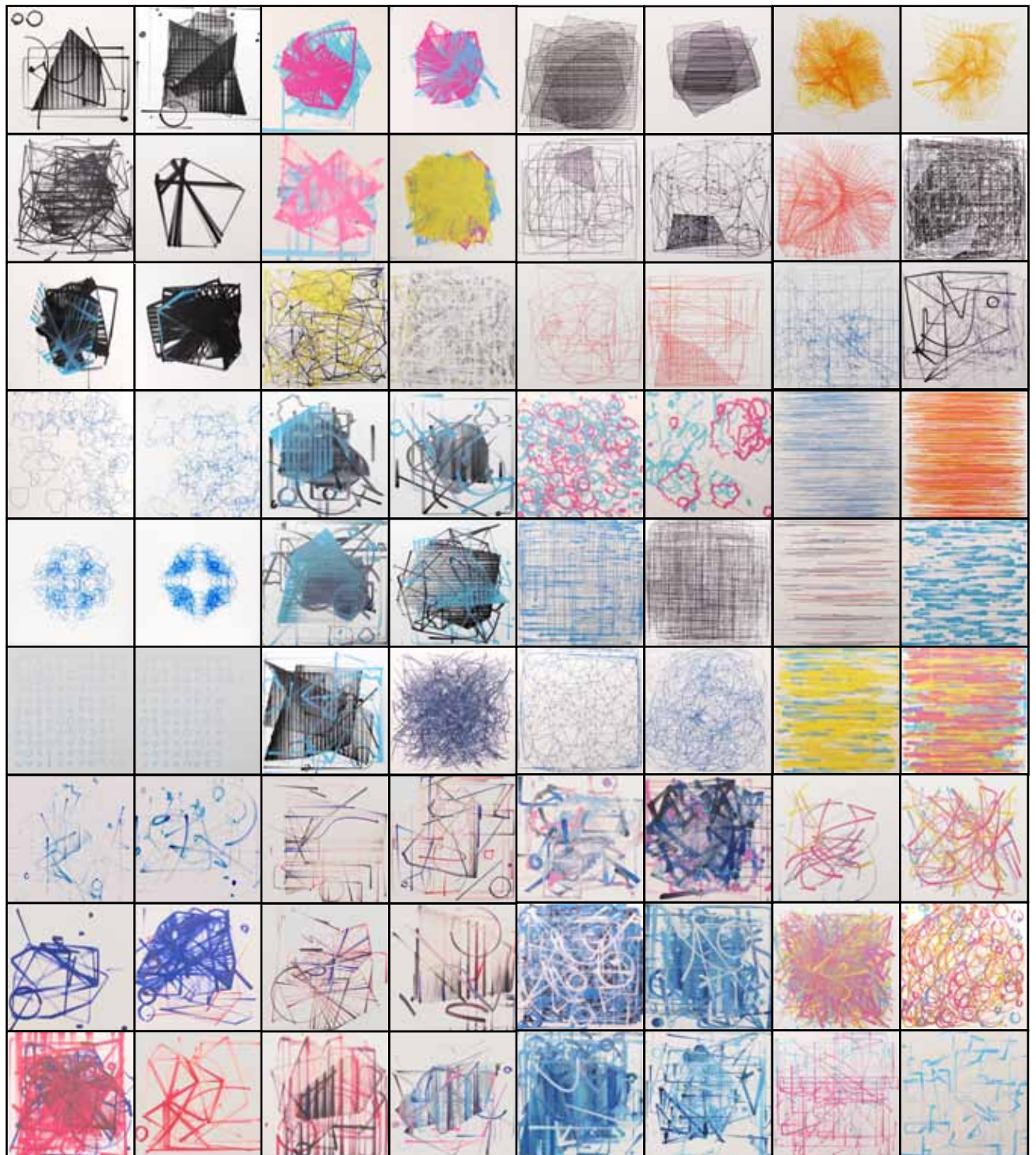
Postrzegam swoją pracę jako całość, na którą składają się zarówno powstałe obrazy, jak i sam proces przeprowadzonego eksperymentu. Konkluzja, do której dążę, to możliwość stanięcia przed jednostkowym dziełem stworzonym na bazie mojego algorytmu i autorskich technologii, i nie dostrzeżenie mojego w nim udziału, może poza świadomością rzucenia pierwszego elementu kuli śnieżowej.

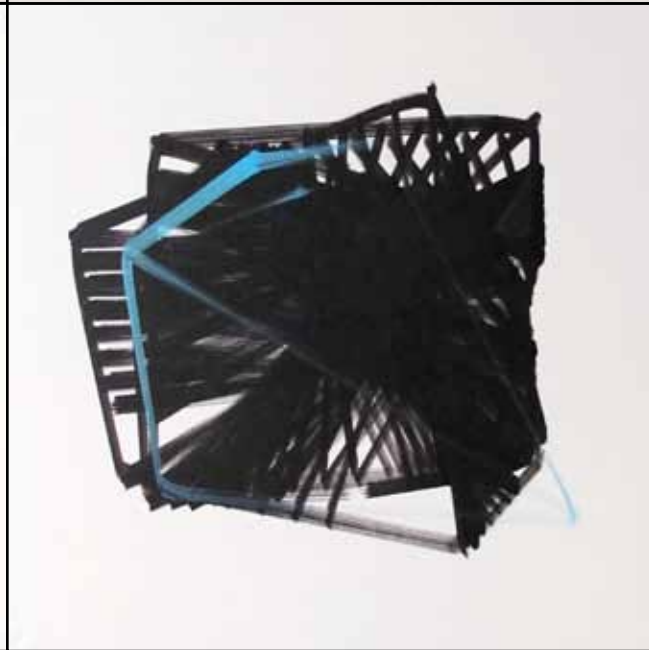
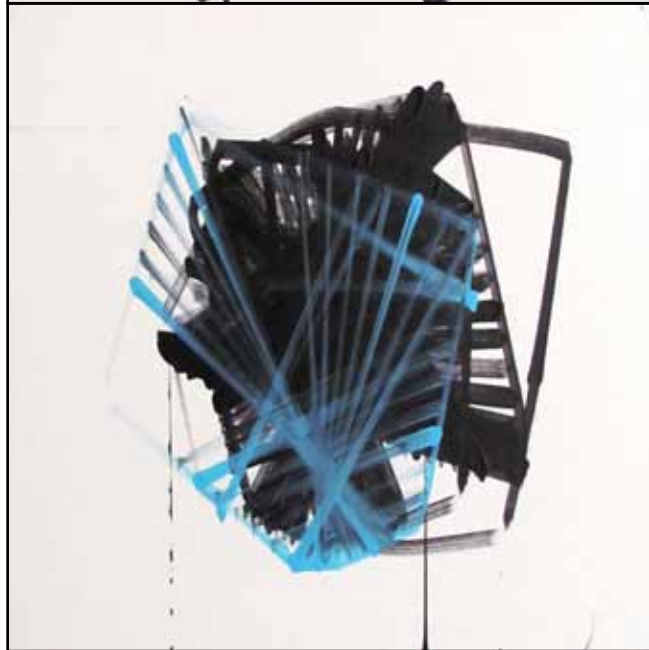
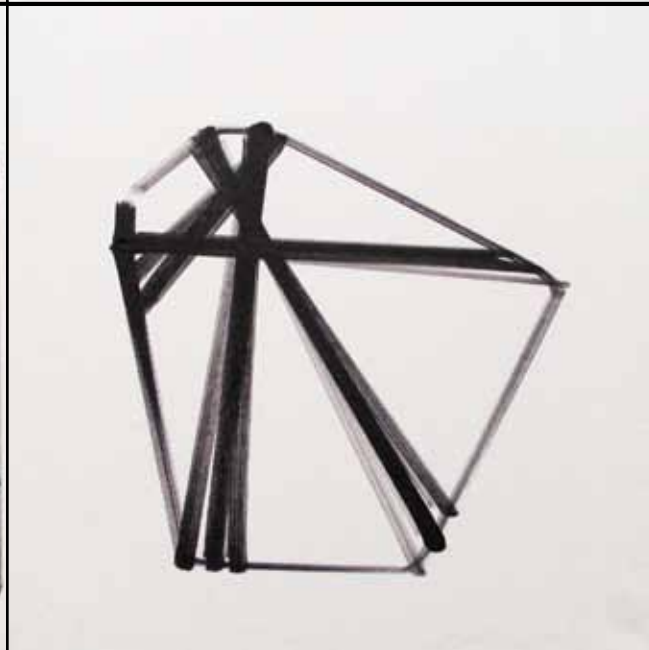
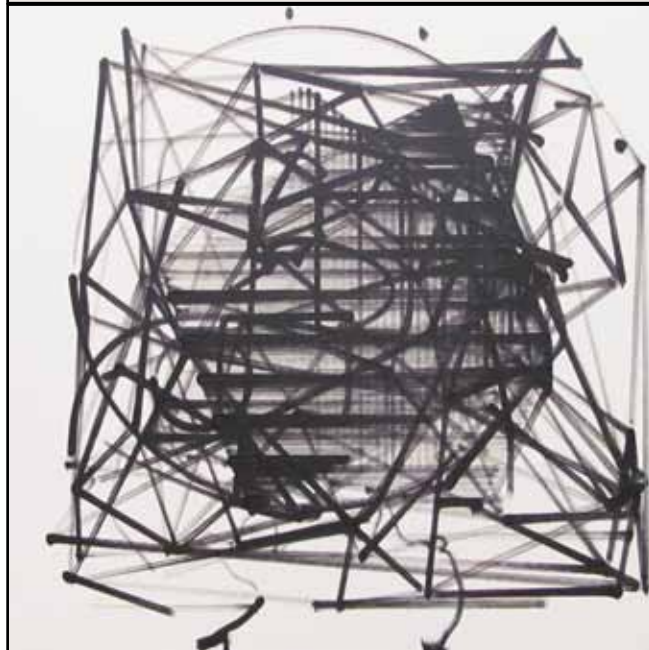
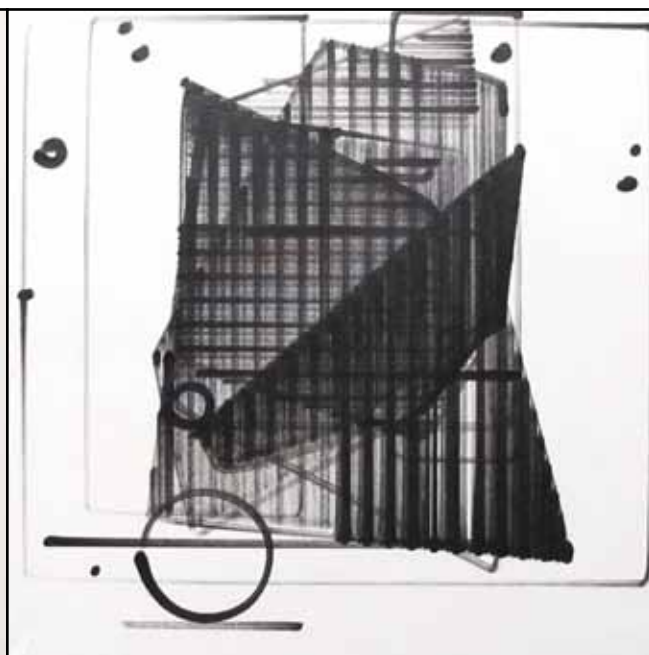
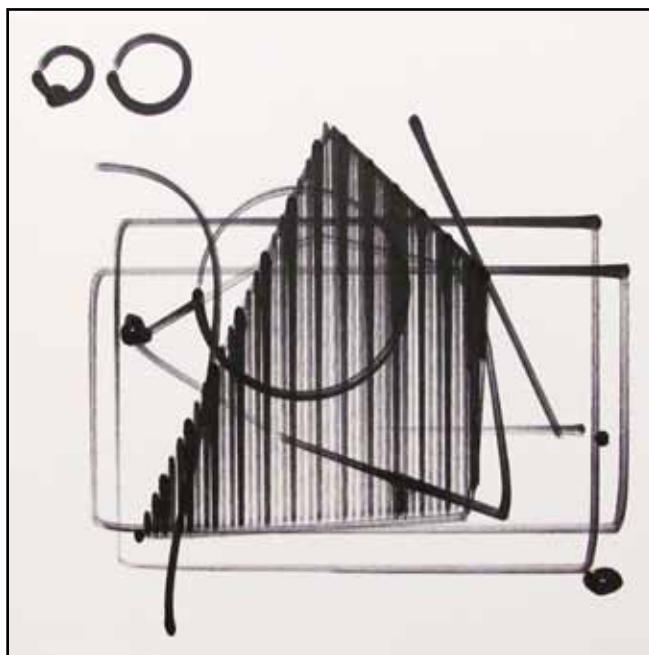
STRESZCZENIE

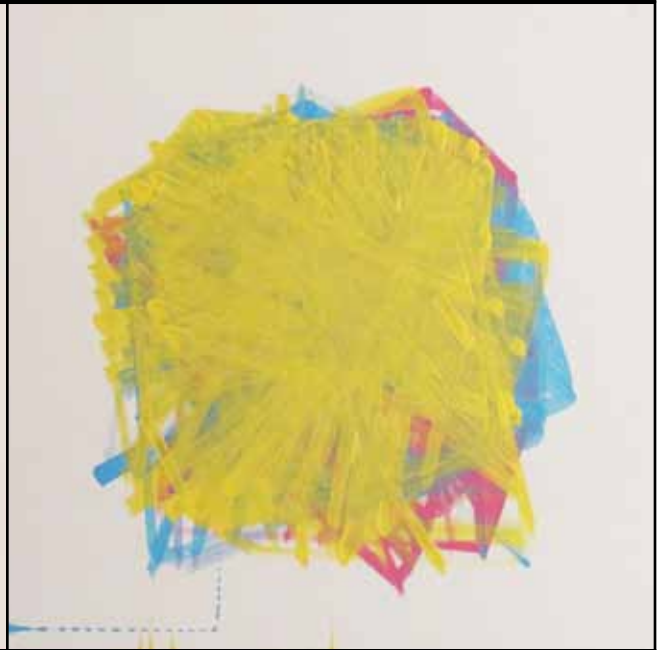
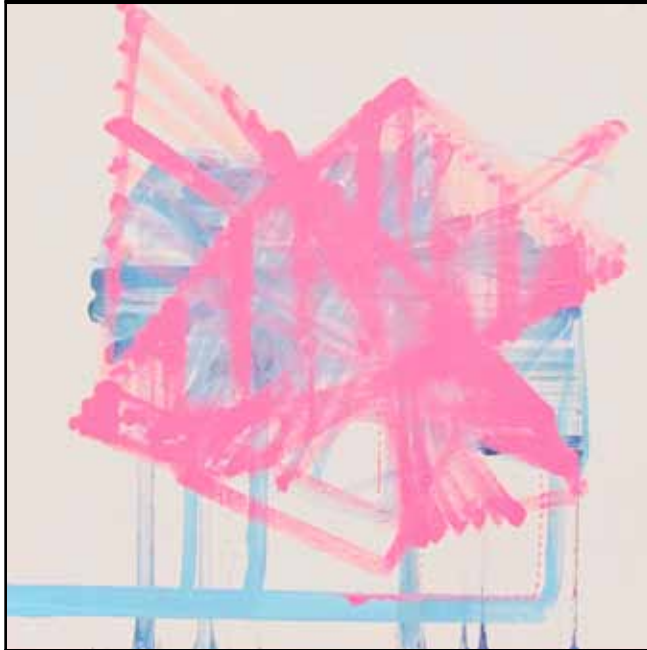
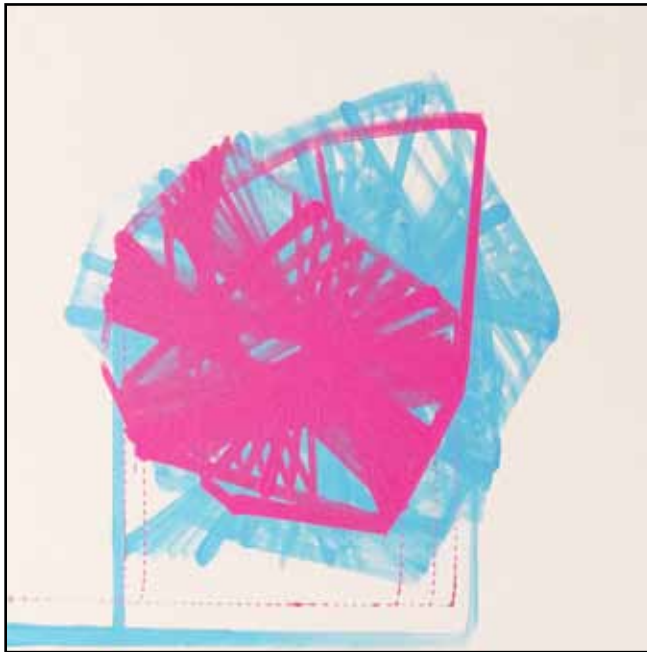
Moja praca jest opisem praktycznego eksperymentu z zakresu malarstwa, prawdopodobieństwa oraz robotyki w ujęciu dzieła sztuki. Opisuję w niej zastosowanie terminów i technologii zaczerpniętych z dziedziny nauk ścisłych jak algorytm, wzór, procedura i ich zastosowania w roli medium artystycznego. Dużą część w mojej pracy zajmują rozważania na temat losowości i prawdopodobieństwa jako, że główną ideą eksperymentu jest stworzenie systemu ograniczającego decyzyjność artysty w powstawaniu dzieła z użyciem tradycyjnych mediów malarskich. W tym przypadku twórczy proces decyzyjny przejmują za człowieka obliczeniowe mechanizmy generujące zdarzenia pseudolosowe. W swoich dociekaniach na temat prawdopodobieństwa proponuję zastąpienie terminu losowości, w odniesieniu do działań artystycznych z użyciem technologii komputerowych, określeniem zdarzeń o pewnym stopniu nieprzewidywalności, który to stopień możemy dookreślić ogólnym stopniowaniem jak wysoki czy niski. Kolejnym ważnym teoretycznym elementem mojej pracy, jest opis terminu algorytmu i jego zastosowania w działaniach artystycznych i różnic w rozumieniu tego terminu w ujęciu matematycznym czy informatycznym. Wymieniam też prosty podział zastosowania algorytmu w sztuce na Algorytm jako narzędzie, Algorytm jako instrukcja, oraz Algorytm jako dzieło. Opisuję także zagadnienia wzoru i proceduralności i ich udziału w stworzeniu mojej pracy doktorskiej.

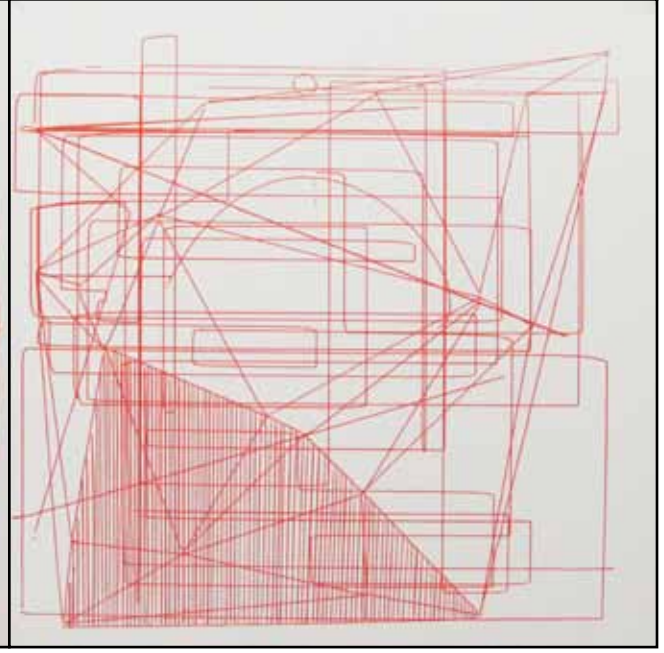
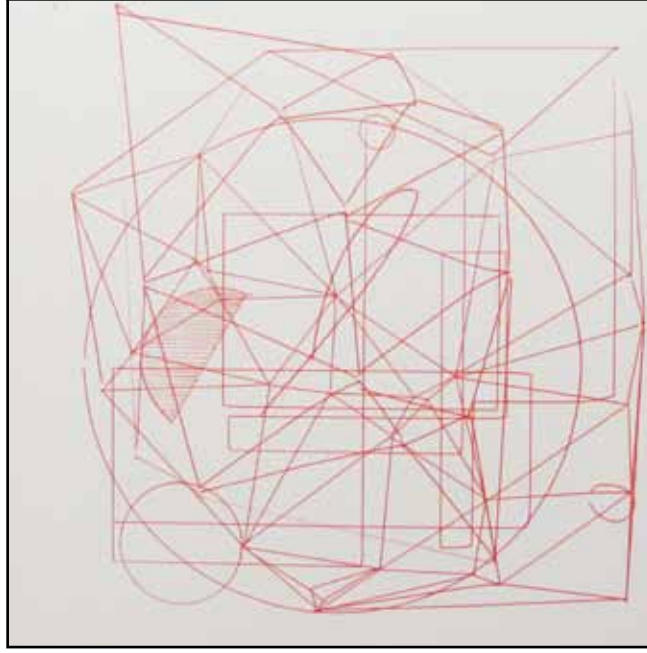
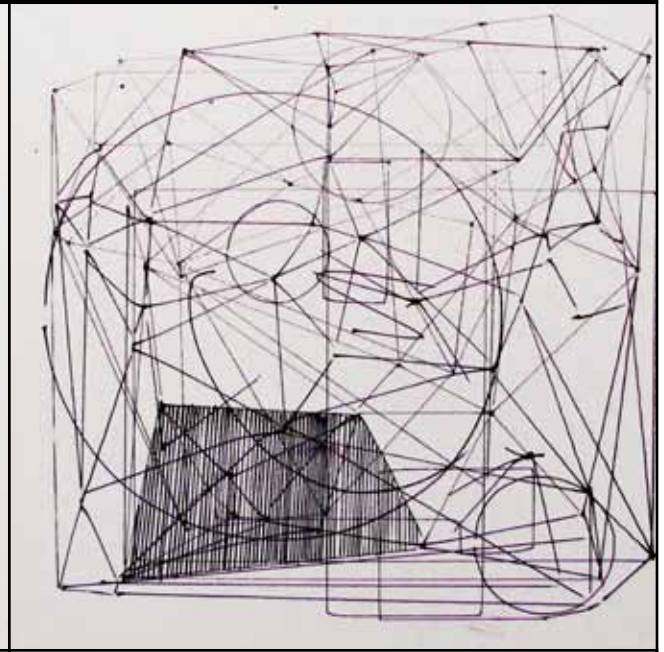
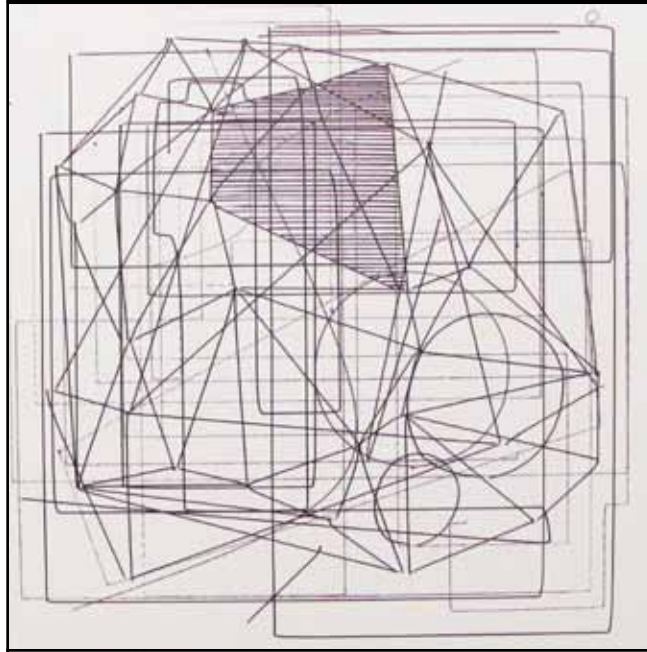
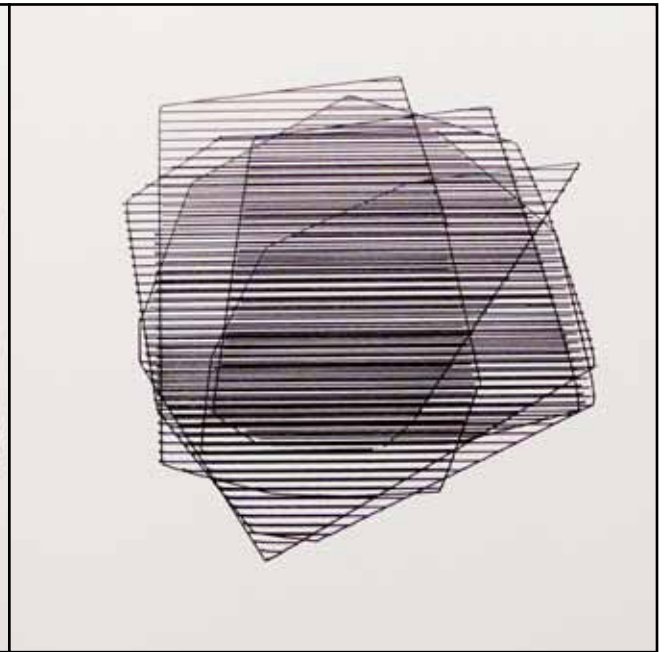
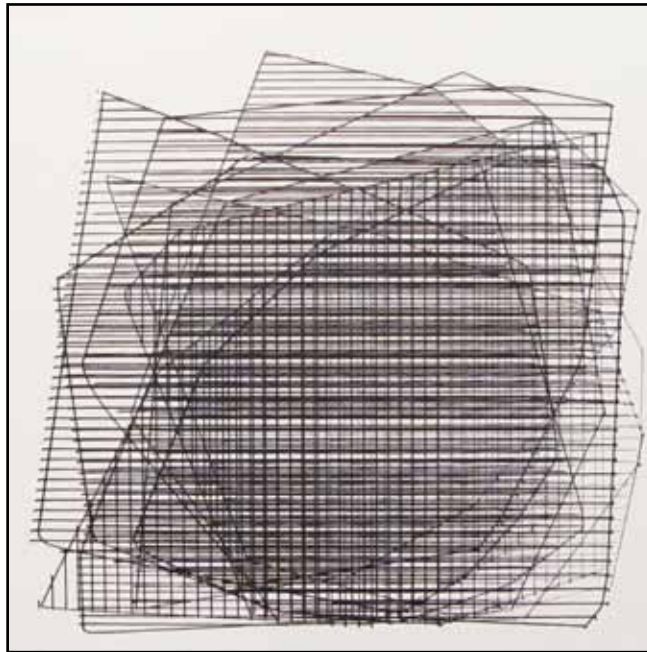
W części praktycznej moja praca to zbudowany od podstaw przeze mnie robot CNC o wymiarach około 100 X 70 cm, posiadający możliwość malowania na powierzchniach płaskich z użyciem tradycyjnych mediów malarskich i rysunkowych, jak pędzel i farba czy pisaki i flamastry różnego rodzaju. Maszyna sterowana jest oprogramowaniem komputerowym mojego autorstwa, generującym podstawowe i skomplikowane kształty geometryczne w oparciu o wzory matematyczne z dziedziny geometrii obliczeniowej. Kompozycje tworzone są przez algorytmy oparte na prostych liczbach pseudolosowych generowanych komputerowo. Wynikiem pracy maszyny i sterującego nią oprogramowania są 72 obrazy o wymiarach 50 X 50 cm tworzące cykl malarski będący podstawą mojego doktoratu. Obserwując działanie algorytmów i pracę maszyny oraz jej efekt w postaci kompozycji malarskich o dużym stopniu nieprzewidywalności, wysuwam konkluzję, w której dzięki nowym technologiom w roli medium artystycznego, dostrzegam możliwość stanięcia przed jednostkowym dziełem stworzonym na bazie mojego algorytmu i autorskich technologii, i nie dostrzeżenie mojego w nim udziału, może poza świadomością rzucenia pierwszego elementu kuli śnieżnej.

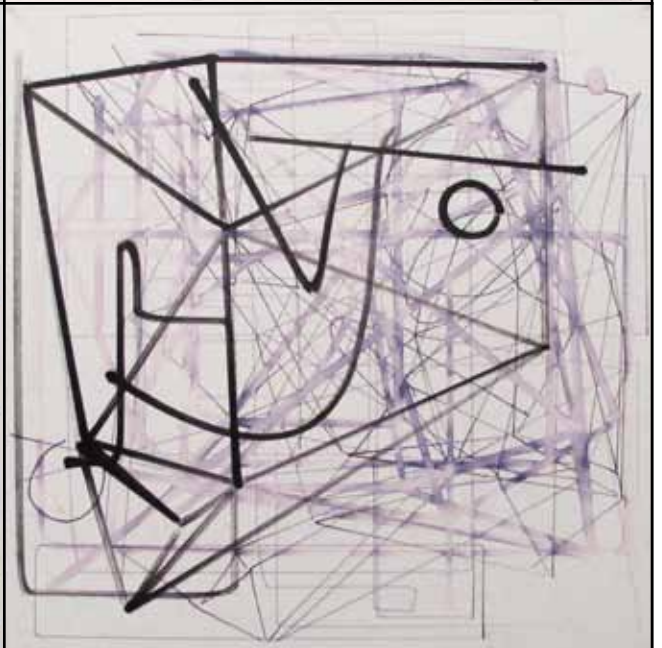
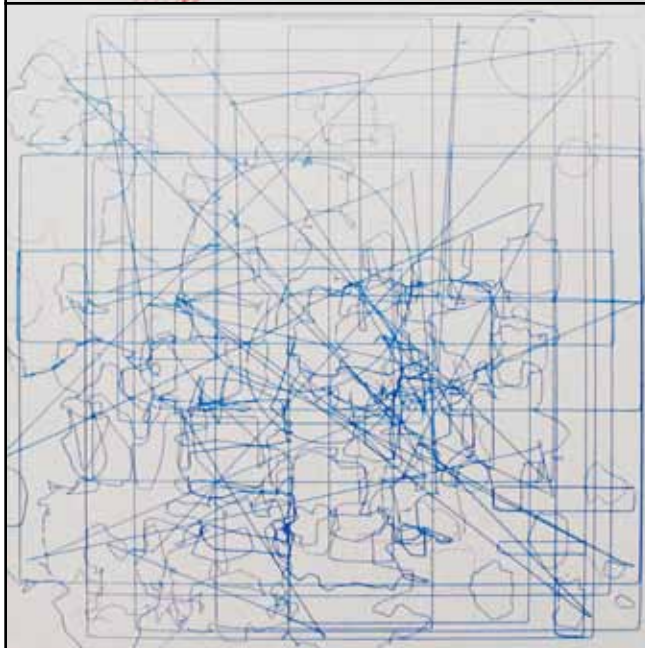
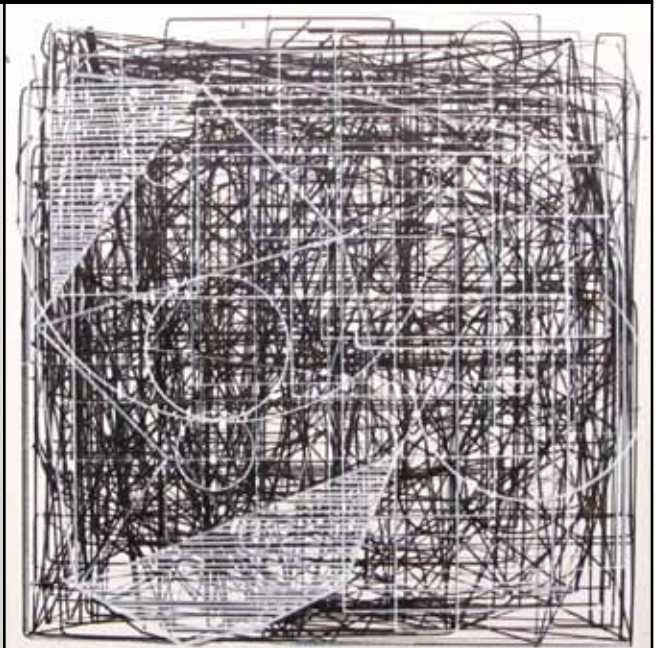
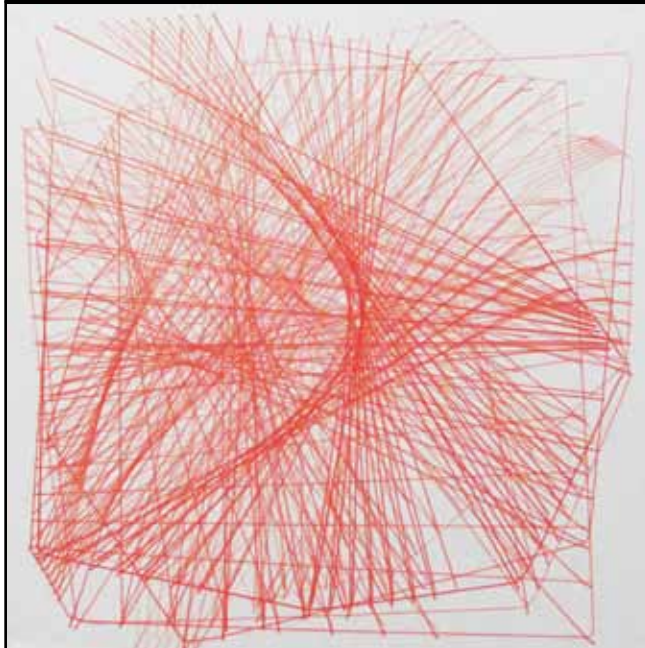
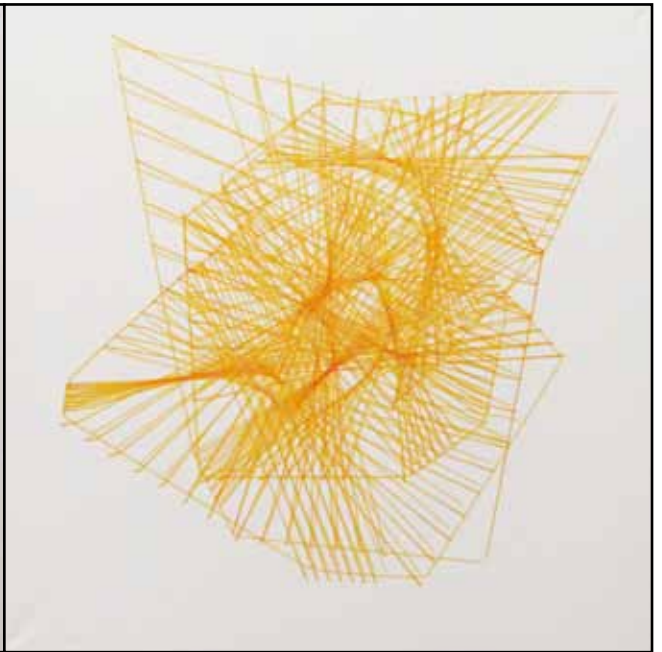
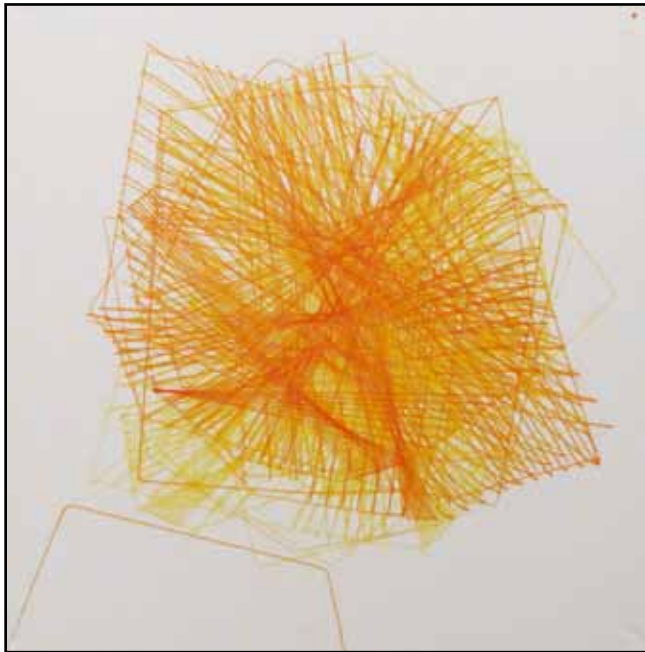
REPRODUKCIJE PRAC

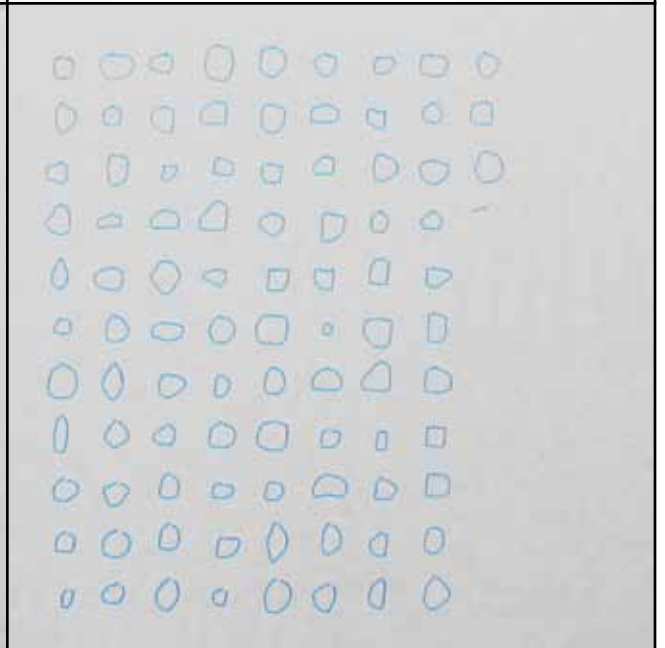
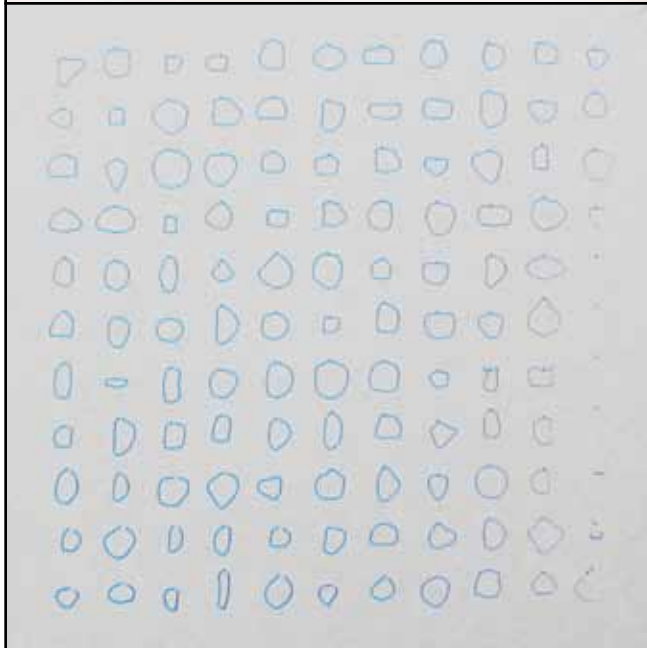
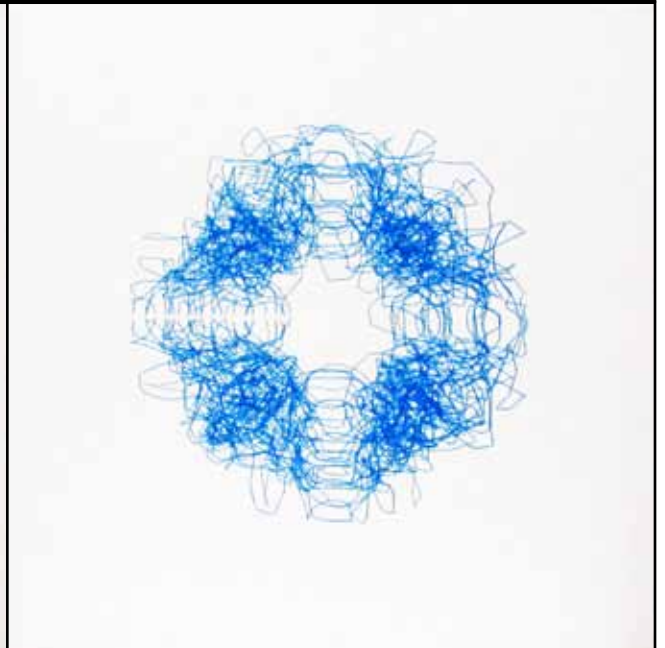
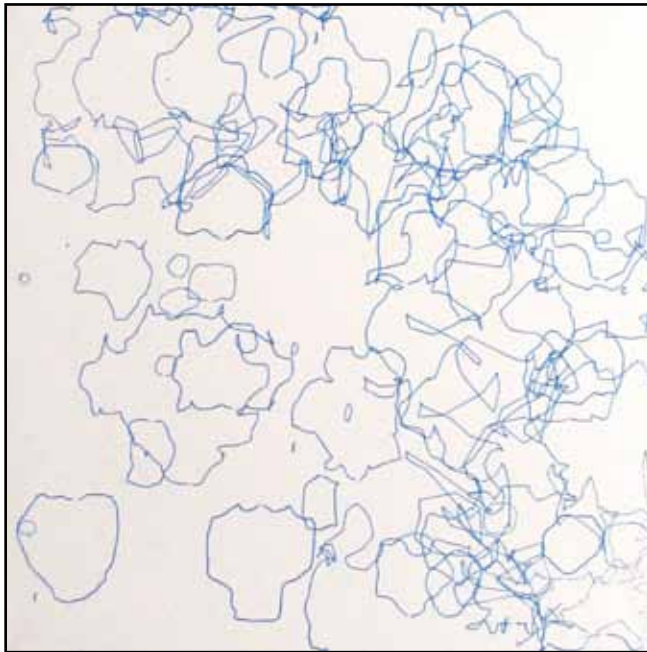


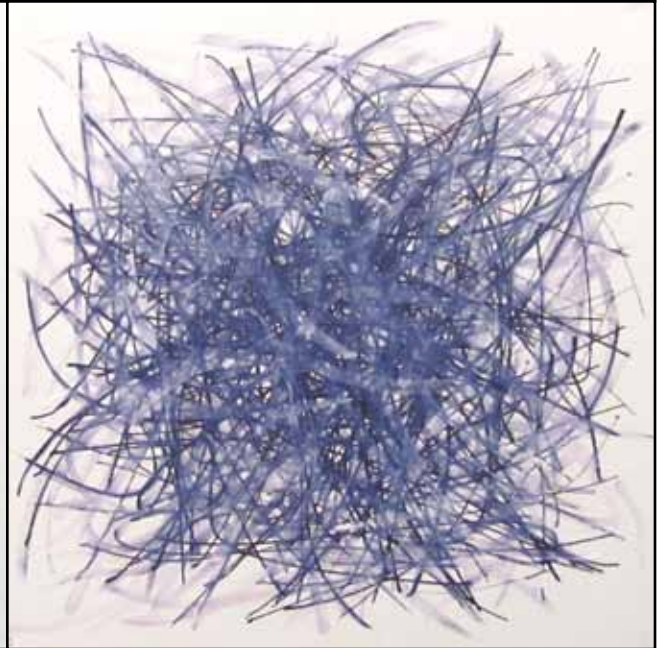
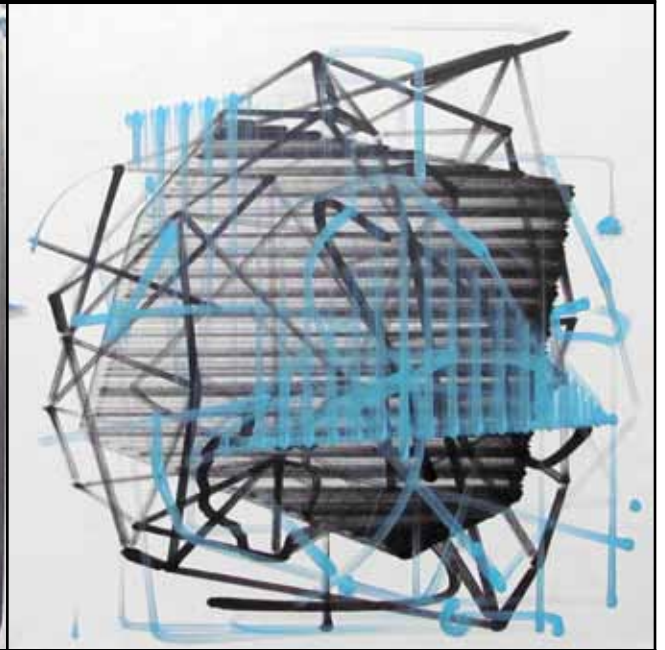
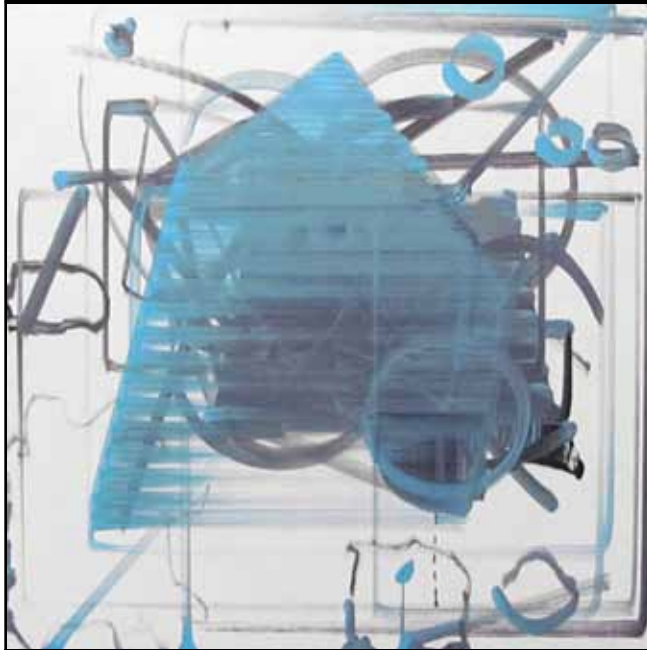
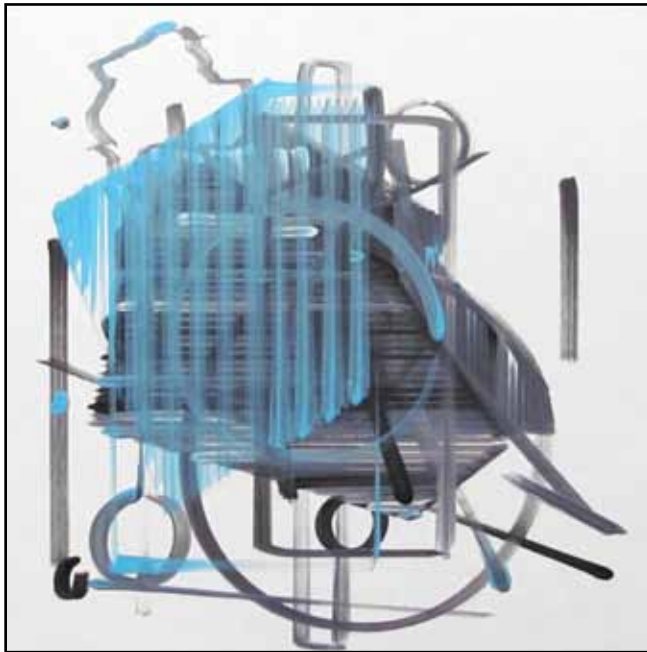


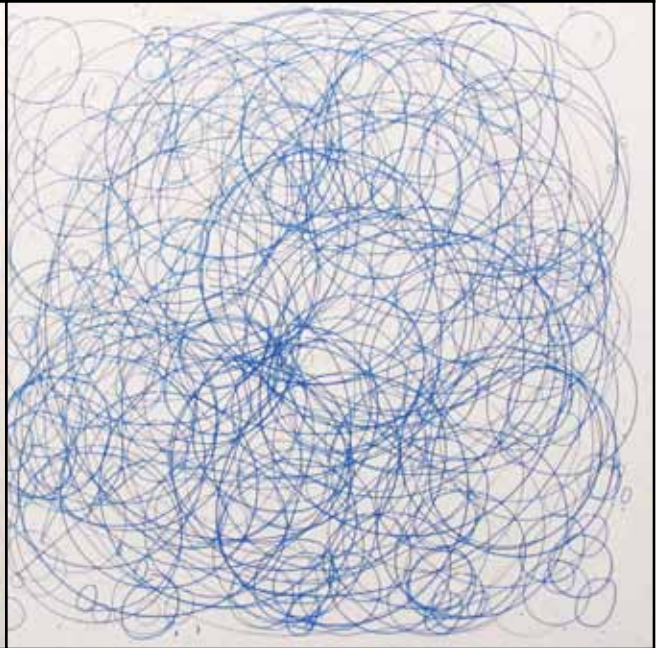
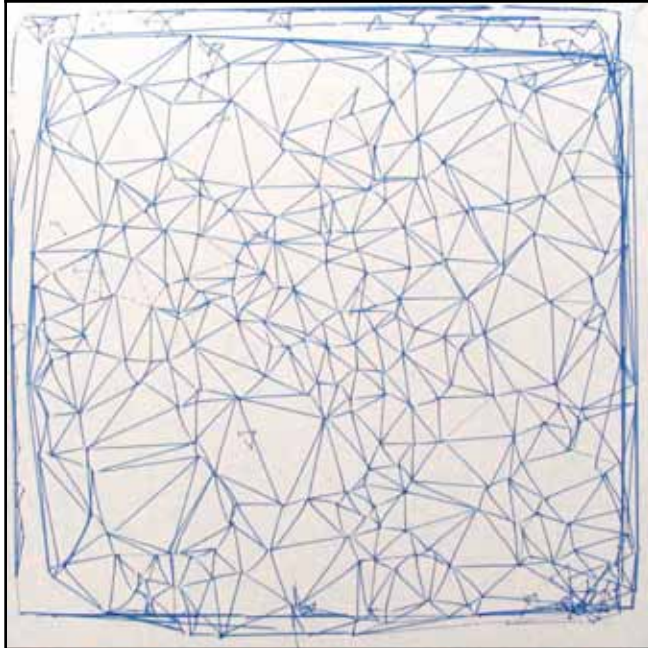
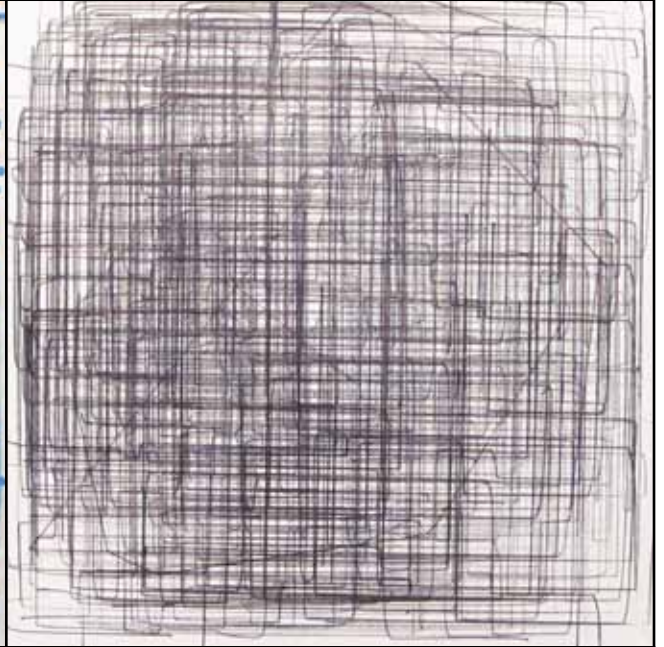
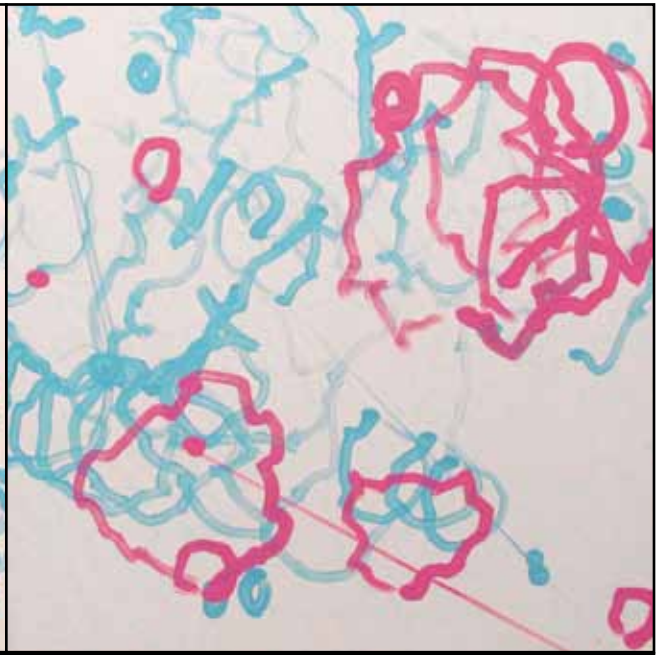
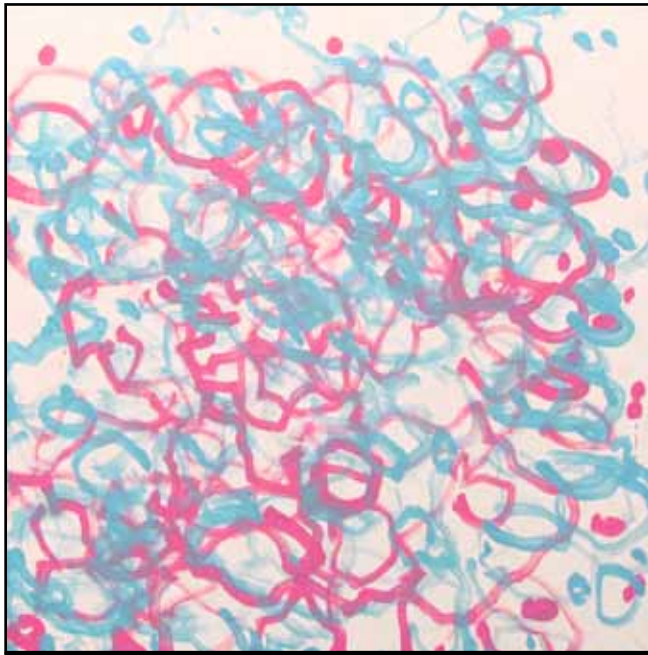


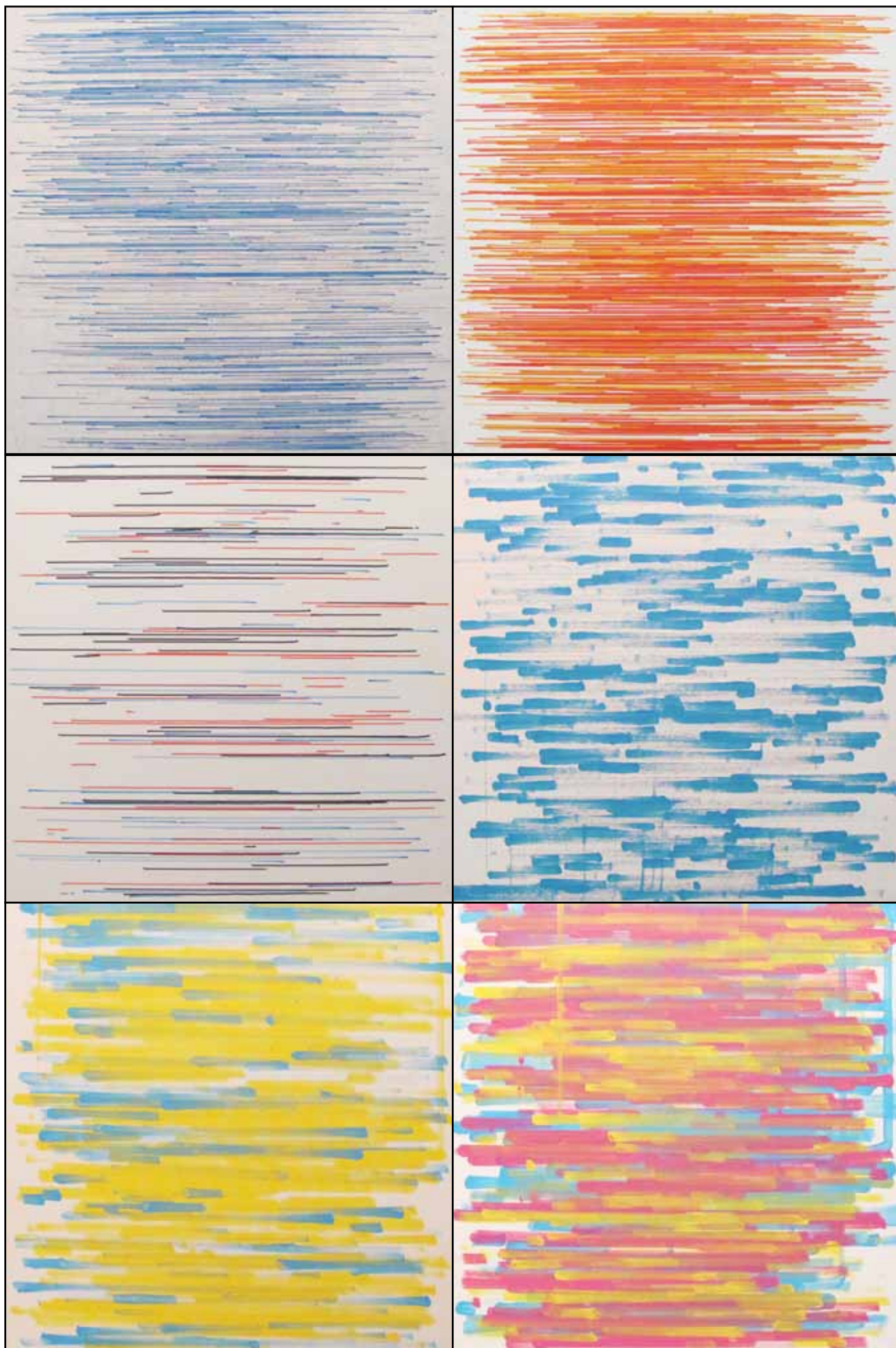


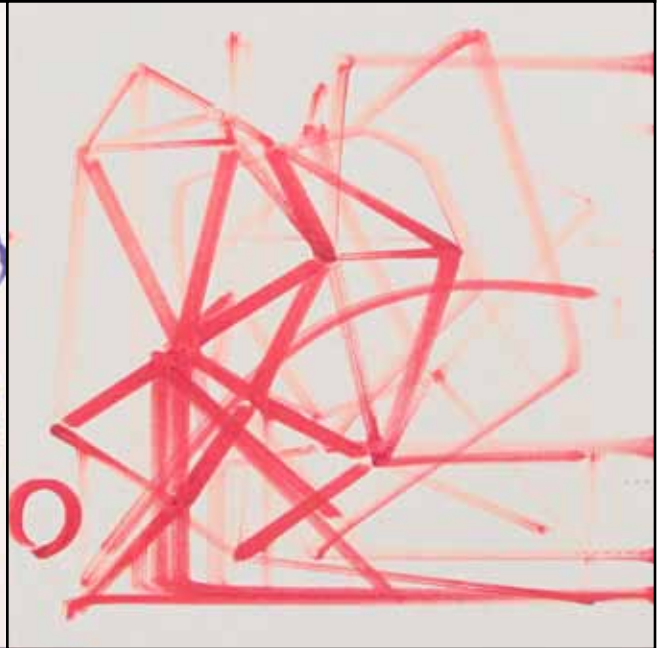
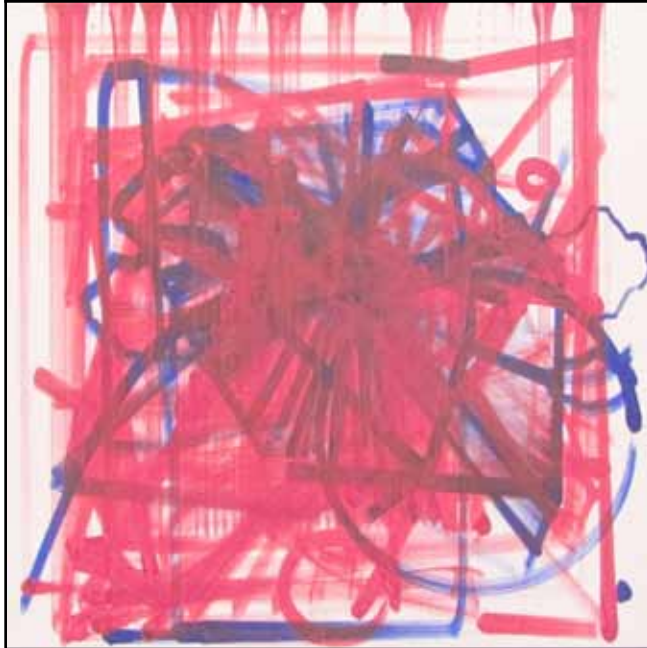
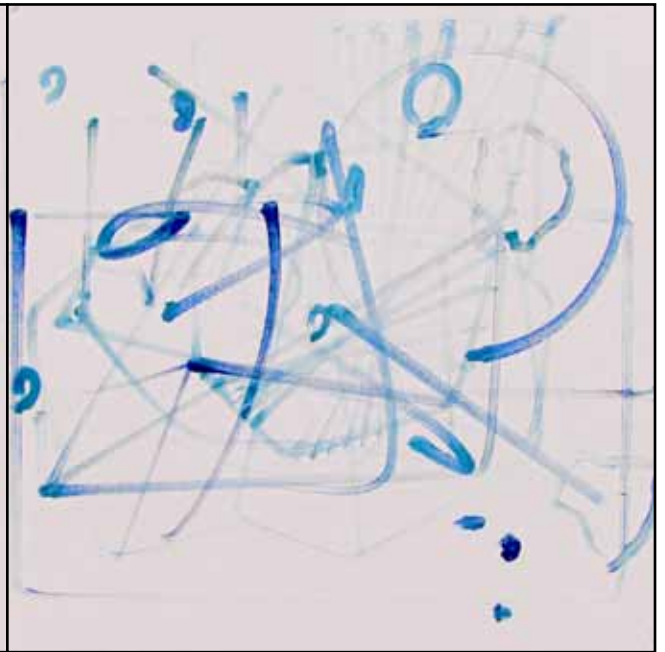
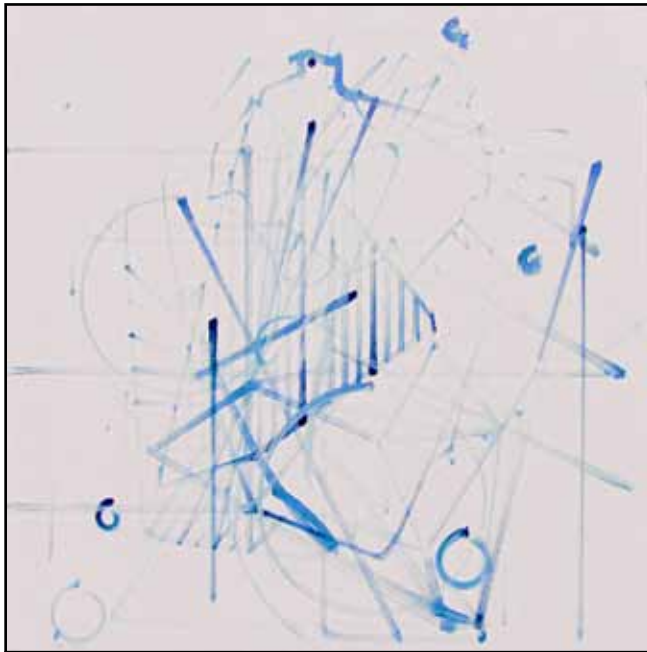


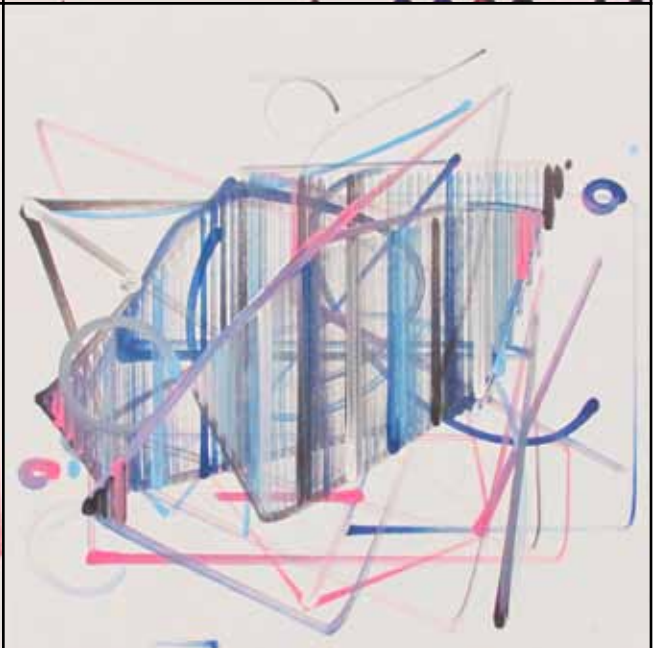
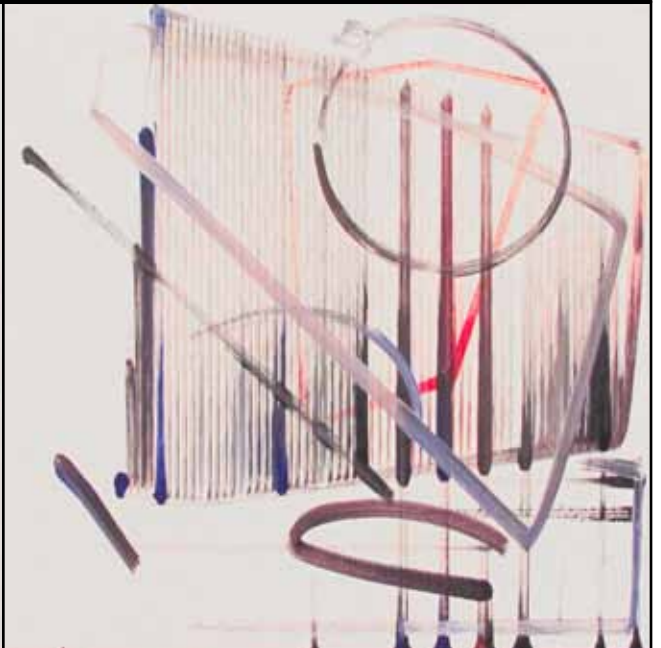
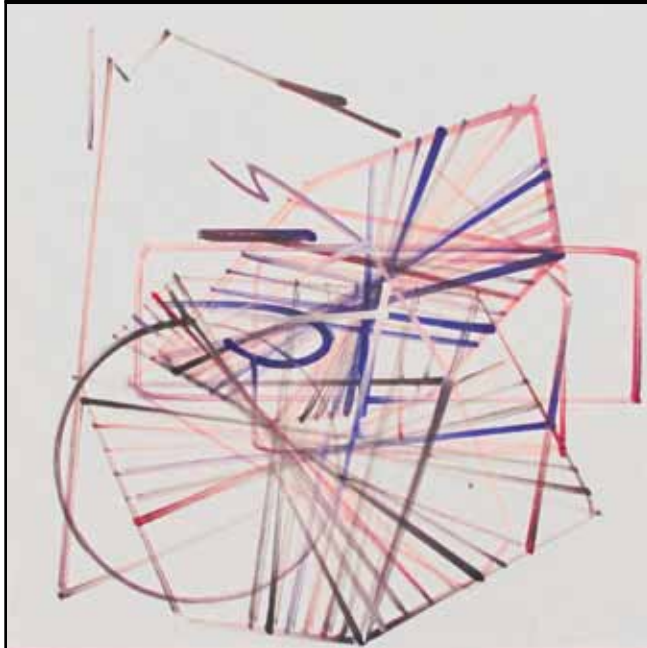
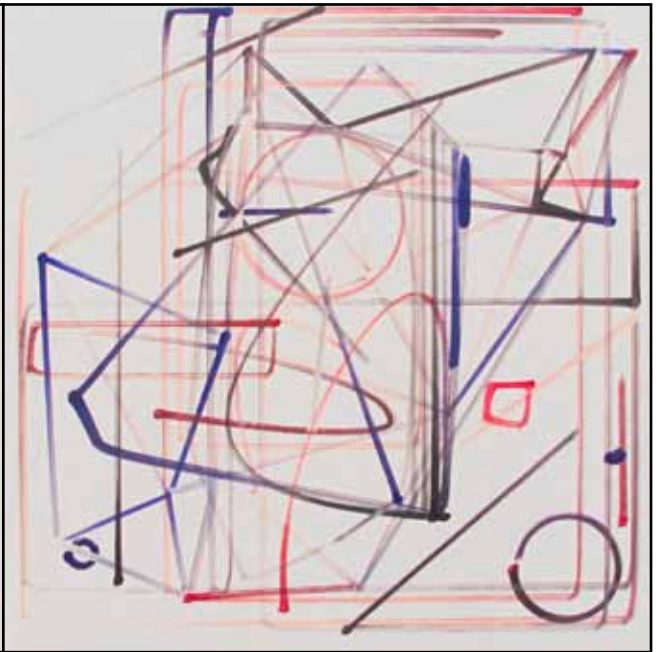
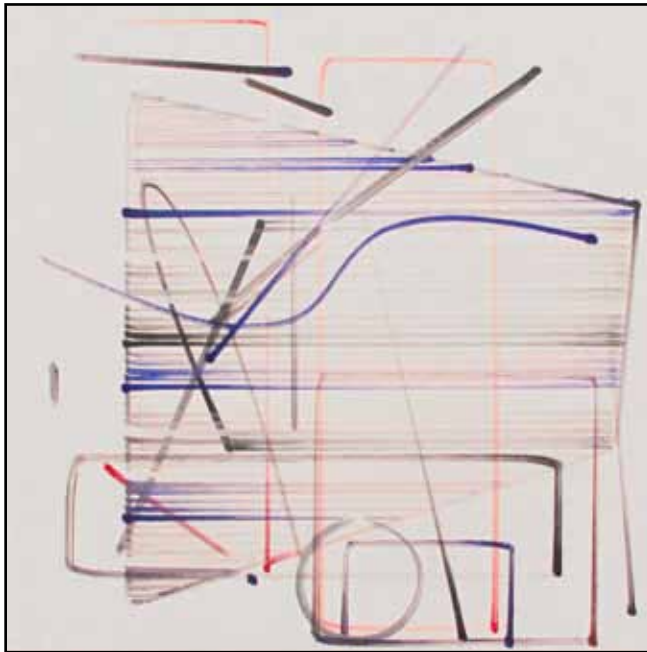


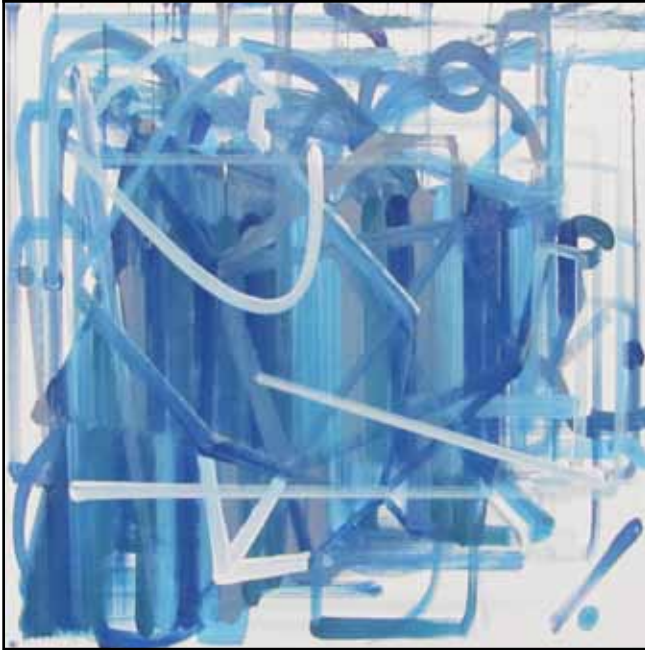
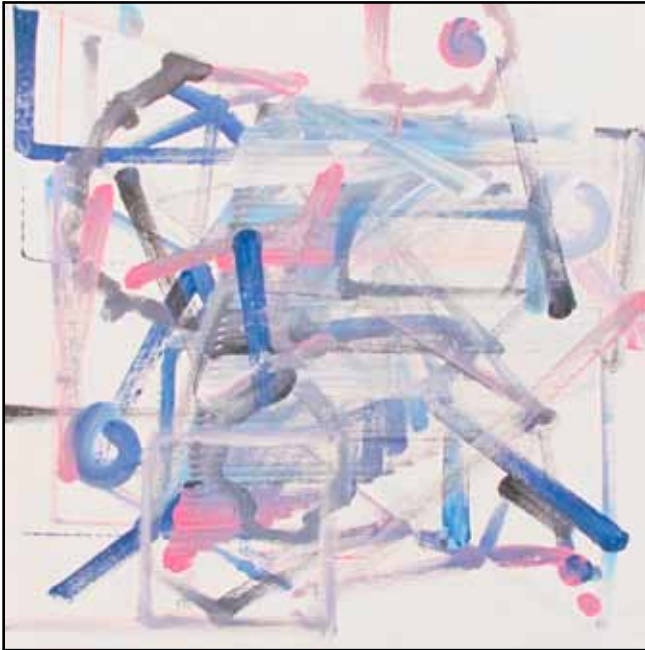


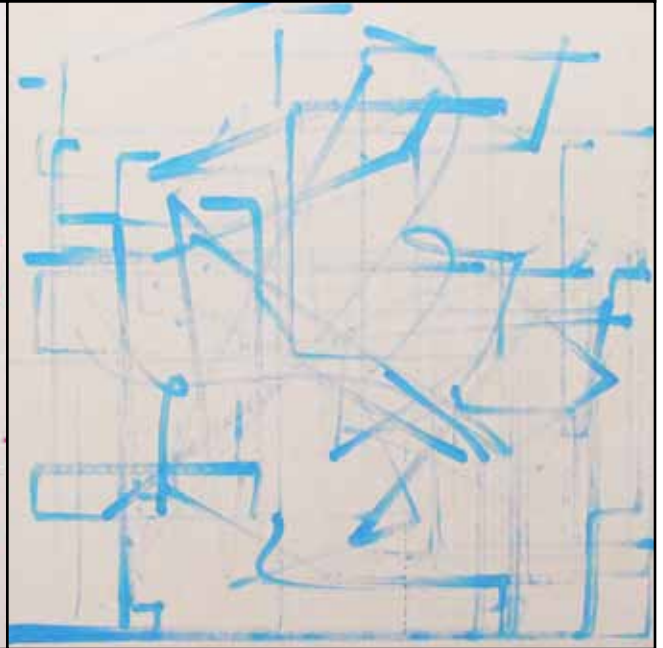
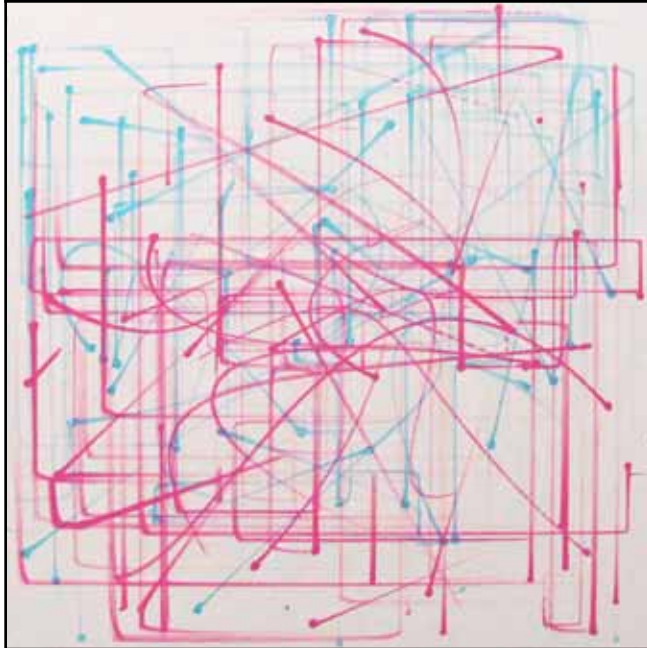
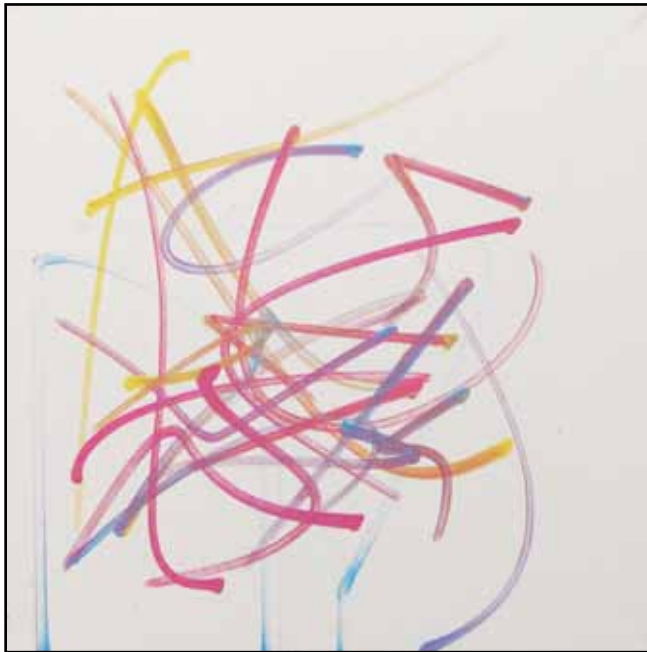












ENGLISH DESCRIPTION OF THE DOCTORAL DISSERTATION

INTRODUCTION

I am fully aware of the fact that my doctoral dissertation may significantly differ from a conventional form of discussing as well as interpreting artistic expression and output since it does not entirely focus on ideas directly related to art or traditional painting techniques. Instead, it explores issues concerning computer and electronic technology, delves into the world of physics, computer science, and mathematics. However, it would be a hopeless undertaking to explain the artistic and conceptual premises of my dissertation without even briefly discussing the mechanism and technologies which, in fact, are applied as a medium in my artistic work. I do not refer to thinkers in the fields sociology, psychology, philosophy, or even social cybernetics, neither do I discuss sociological aspects of my works as I do not address social issues. I do not transform images of reality graphically. Being the author of the procedure and algorithm, I search for the graphic diversity in the consecutive iterations of a formula generating a procedural image within the limits of the algorithm performance. My studio is a laboratory and the process of creating a work of art is an experiment characterised by numerous iterations of the procedure generating a painting. The subject of my research concerns the performance of a highly unpredictable computer algorithm generating a graphic image as well as displaying it by means of output devices such as a television screen, computer projector, or, finally, a machine which produces the formula using conventional painting tools. Even though one can link the idea of using a painting machine with Henry Ford's conveyor-belt assembly line in the context of art as well as broadly understood George Ritzer's concept of McDonal-dization, both associations fall outside the scope of my dissertation.

First and foremost, the aim of my work is to introduce the concepts related to the medium I use such as an algorithm, computer code, randomness, and proceduralism regarding art. Exploring the meaning of the above terms will allow me to discuss conceptual premises of my artistic output and ideas which have influenced my views on art.

Although I can decide to avoid discussing philosophical or sociological issues, it is vital to mention artistic activities of a similar nature taking into account the medium, assumptions, or conceptual sphere as well as investigating their origins. The idea of applying computer techniques in artistic or graphic productions is more than a half century old and has been utilized by a great number of recognized artists in a variety of fields and even more amateur art enthusiasts and hobbyists.

It is a relatively easy task to trace the origins of computer art as they are closely connected to the development of its main medium – analogue and digital electronics as well as electronic computing machines. The vast majority of sources point to the image displayed by the Semi-Automatic Ground Environment (SAGE), a computer that coordinated the USA radar system in the early 1950s. Scientists and programmers, who supervised computer operations, programmed it by means of punched cards to display a copy of George Petty's popular drawing depicting a scantily clad pin-up girl on a archaic, compared to modern ones, screen using vector lines.

Even though it certainly was not the first image generated and displayed electronically, it is safe to assume that it was one of the first, if not the first, attempt to figuratively produce a computer graphic image of a human body in considerable detail. I consider this example an extraordinary form of prediction for future generations, dating back to the beginning of the computing machine era. The technology worth tens of millions of dollars, developed to shelter humanity from a nuclear holocaust, is transformed by a programmer into a medium for pop culture erotic content, bringing to mind associations with an impressive variety of applications of modern technologies, for example the Internet, and their actual use.

Despite being sociological by nature, the example above leads to an inescapable conclusion that computer technologies had belonged to a scientific world for a considerable amount of time. Due to the lack of user-friendly interface or input and output equipment, the proces-

sing power of a computer was in the hands of scientists who understood its programming and algorithm languages. Many years had passed before Palo Alto Research Centre (formerly Xerox PARC) laboratory, in 1970s, provided an average user with interface solutions facilitating work with a computer. One was able to witness the development and popularization of a personal computer (PC) to such an extent that it has become an integral element of an average household. Early 8-bit computers with a simplified version of programming languages for beginners (BASIC) allowed computer enthusiasts and hobbyists to take control over the processing power. The expert knowledge of complicated machine code was no longer required as it was possible to operate a computer by means of simple English-language commands. Consequently, an average computer user could observe graphic effects of applying code on a monitor or, more often, television screen.

My first Atari computer still brings back fond memories of an eight-year-old boy who copies the first lines of code from a German programming textbook although he couldn't understand a word. Having executed the program, to my great astonishment, a real lightning appeared on my television screen. Today, I am aware of the fact that it was just a depiction of an ordinary white broken line, composed of clearly visible pixels, on a black background. At the time, however, I considered it a true force of nature which I tamed with the help of a system of alphanumeric characters. What is more, each time the program was executed, the lightning changed its shape and appeared in different sections of the screen. Before I had discovered the answer by improving my knowledge of computer-generated random numbers, the idea was virtually incomprehensible to me, even magical. Nevertheless, before an ordinary computer user could use a keyboard to create first computer-generated patterns, someone had had to devise mathematical formulas defining graphic forms and displaying them on archaic oscilloscope screens by means of analogue electronics. At the time, their works were already recognized as art, or at least the germ of computer art.

PIONEERS, ARTISTS, INSPIRATIONS

One of the first creators of computer graphics was Ben Laposky who displayed his works in the form of oscilloscope projections transferred into photosensitive paper called 'Oscillons'. His research in the field of automatically controlled abstract organic patterns, which he initiated as early as the late 1940s, evolved into an art form showed on his solo exhibition entitled *Electronic Abstractions*, held by Stanford Museum in Cherokee, Iowa in 1952.

Another important figure in the field of computer art was Herbert W. Franke, an Austrian scientist, physicist, science-fiction writer, and artist. In his artistic works, he also utilized an electronic oscilloscope image and photographic technologies as the medium for displaying it. In addition, Franke wrote several influential publications on computer art, such as the book *Computer Graphics: Computer Art* (1971) or essays: *The Influence of Computer Graphics on Art and Society* and *The Expanding Medium: The Future of Computer Art* in which he predicts further development of real-time computer-generated animation accurately. Despite the significant achievements of the above-mentioned creators, one cannot forget to give credit to Roy Ascott for his important contribution to the development and popularization of computer art. Since the early 1960s, his reflections on computer art, especially interactive computer art, have been groundbreaking. In the context of my dissertation, the most inspiring are Ascott's ideas on computer and electronic media as tools, included in *Behaviourist Art and the Cybernetic Vision*. Ascott does not perceive a computer as 'a physical tool in the sense that an aluminium casting plant or welding gear are tools – that is, extensions of physical power.' He treats a computer as an instrument which develops thinking abilities and an intelligence booster. From my point of view, it is also a perfect medium for spreading ideas and conceptual form.

The 1960s were unquestionably an important decade for the beginnings of computer art and its future variants. 1965 was especially significant because of three important exhibitions dedicated to this trend. Firstly, Georg Nees held his *Computer Graphik* exhibition at the University of Stuttgart. Studying mathematics and physics, among other things, he participated in the work on the construction of plotting machines. The experience helped him in producing his artistic works by utilizing programmed drawing machines and mathematical formulas to generate geometrical patterns. The second exhibition entitled *Computer-Generated Pictures* at the Howard Wise Gallery in New York showed the works of Michael Noll and Bela Julesz. Last but not least, Galerie Wendelin Niedlich in Stuttgart hosted the exhibition devoted to works of Frieder Nake and, once more, Georg Nees. *Computer-Grafik* (Nake & Nees) included pieces of art produced by computer-controlled drawing machines.

These artistic events and their wide critical acclaim not only contributed to the popularization of computer technologies in art but also, what is more important in the context of my work, popularized the works produced by means of plotters or CNC machines capable of drawing, painting, or even sculpting. Obviously, since then, the history of computer art has quoted numerous examples of artists expanding the trend for works created by means of computer programming or computational intelligence, for instance Harold Cohen, whose computer program AARON produces images autonomously utilizing both a screen and drawing machine, or Pindar Van Arman, who uses drawing robots as the creation medium for his art. His computer system *Cloudpainter* appears to be even more sophisticated than AARON.

Furthermore, systematising the events and trends concerning computer art shows attempts even before 1965. It is worth mentioning drawing by means of machines in Tinguely's kinetic sculptures or Desmond Paul Henry's drawing machines powered not only kinetically but also by analogue electronics, shown on the exhibition *Ideographs* in London, 1962.

Quoting all the representative examples of this art form is far beyond my needs. I only wish to emphasize the fact that I am fully aware of the existence of both pioneering and contemporary works in this field and their inspiring contribution to my artistic output.

RANDOMNESS

In my view, there are no chance events in real life. The position of every atom in the universe is determined by countless conditions and forces which have decided the shape of the world we know. Their arrangement has never been random since all the components of matter rearrange their positions in the infinite causal chain as if billiard balls were hit by the Big Bang. The phenomenon one can perceive as random is actually affected by variables with a high degree of unpredictability. An efficient computational system and accurate data would allow to retrace the route each grain of sand took before finding itself under one's feet as it would be possible to examine the natural phenomena of water currents, wind power, rock erosion, or terraforming processes. Although the effect seems to be random, even chaotic, the causes are always the result of precisely defined principles of matter existence.

Randomness has inspired humanity for ages. History tells about daredevils who tempted fate by placing a single round in a revolver and spinning the cylinder, rulers who determined the fate of their subjects on the basis of a throw of dice, stones or strips of wood with prophecies, which we interpreted by 'chosen ones' capable of deciphering and reading the language of the universe. In games, randomness determined the fate of players, their fortune, or even lives. Therefore, it is not surprising that a number of people still find randomness quite attractive. Unpredictability can be perceived as divine and supernatural.

An iterative animation of the appearing and disappearing cursor in a word processor is monotonously predictable and no one seems to be intrigued to observe the pointer jumping between appearing characters, which cover subsequent pages of a document. Similarly, the idea of moving a square on the surface of the computer screen using a keyboard will gain no one's admiration, especially taking into account today's technological development. However, it would be intriguing to see a randomly generated computer animation of the square on the graphic plane of the screen combined with its multiplication which creates the impression of observing unpredictable systems: a simulation of life, a specific representation of an ant farm managed by 'artificial intelligence'. What would happen if one was able to foresee these movements? Would one still discern the influence of fate then? In fact, it is possible to predict them. A considerable number of artists use computer-generated 'random numbers' without realizing that they are not really random. Therefore, computer scientists call them pseudorandom numbers for a good reason. To illustrate the following example, one needs to visualise a simple mathematical algorithm converting any number into another, in the range from 0 to 9. In fact, it is sufficient to calculate the cosine of any number and then its absolute value which gives fractional values in the range from 0 to 1. Next, the number is rounded to one decimal place and multiplied by 10. Consequently, one gets values falling within the defined range. In the majority of cases, giving a random number results in inability to predict the result of this formula. However, the most important in this example is the fact that giving the same number again generates the identical result. I will present the following example using a pseudocode (a simplified version of computer code) which ought to demonstrate clearly how a computer-generated random number algorithm works.

Calculation of value R (random) for any number X

1. Take the value X
// calculate the cosine of the number X and its absolute value
2. If $\cos(X) < 0$ then $R = \cos(X) * -1$
3. If $\cos(X) \geq 0$ then $R = \cos(X)$
// round the number R to one decimal place
4. $R = \text{round}(R)$
// bring the number R to a value from 0 to 10
5. $R = R * 10$

It is an oversimplified example of a pseudorandom number generator. In fact, an advanced system algorithm is capable of generating numbers with even higher degree of unpredictability and meeting the principles of probability theory, however, one rule is invariable: If you replace X with the same number, it always generates the same number R.

At any decisive moment, if X is replaced with the time of the computer's system clock (the current time given in seconds, including the date), the result has a very high degree of unpredictability (additionally, input values will never be repeatable in the subsequent randomizations as the computer's clock does not turn back). However, the values generated simultaneously on two different computers will be identical.

The discussion reveals a proportional relationship between predictability and randomness: the more a phenomenon is predictable, the more elements of the causal chain one is able to reconstruct backwards and less 'enigmatic' or random the outcome seems to be. In the process of pseudorandom number generation, the number on the basis of which the pseudorandom value is generated is called a seed. In the example above, the number of seconds of the system clock becomes the seed. The term will be an important element in terms of the future discussion of randomness and algorithms generating the form of a work of art.

Higher complexity of operation as well as higher degree of unpredictability can be obtained in two ways. Firstly, by using more efficient source of input numbers (seeds) than a system clock which can mean, for example building a seed generator. Secondly, a control algorithm of the end result may have its own set of instructions which responds in a complex manner depending on the randomisation result.

Example:

The artist creates a square black-and-white painting.

A piece of white canvas has been divided into 100 equal squares to form a 10 by 10 grid.

The artist tosses a coin.

If it is heads, the square is painted black.

If it is tails, the square remains white.

The artist chooses the next square and tosses a coin.

Return to step 1.

After a hundred iterations of the algorithm, all the square spaces, line by line, have been filled white or black creating a representation of randomly generated values.

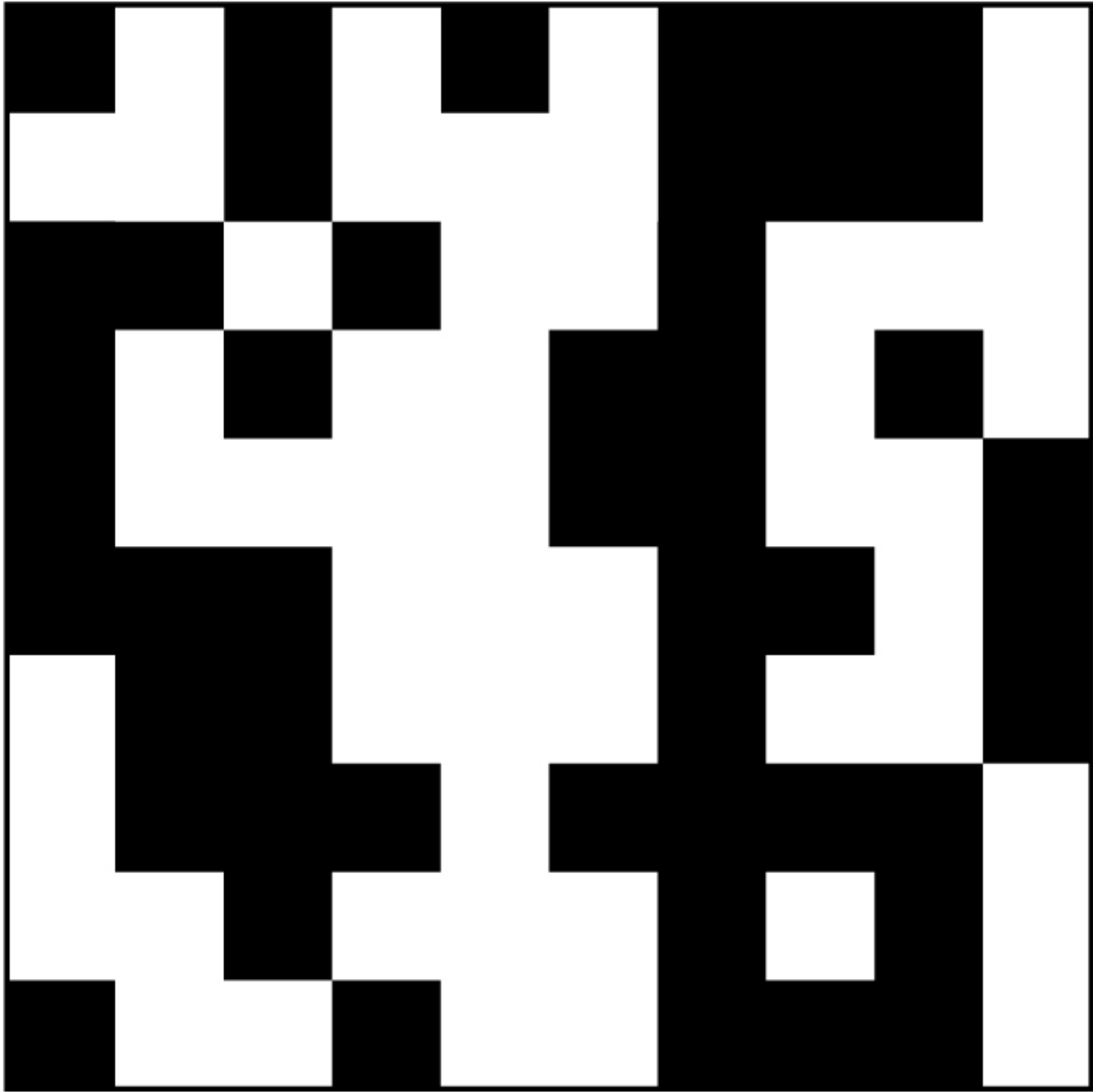
The number of created variations seems to be vast, however, without repetition, it will provide only 10,000 different end results. Although in terms of quantity the number considerably exceeds the life output of many an artist, the formal effect overwhelms by its monotony. In accordance with the principles of probability, filling the whole line with white or black or gaining the predominance of one of the colours is highly unlikely. Furthermore, the majority of results in a line (that is 10 flips of a coin) consist of the comparable number of heads and tails, though in different order. Consequently, the paintings generated in such a way are graphically similar and relatively predictable.

The example of the experiment is shown below.

In order to produce more intriguing graphical effect, it is advisable to increase a degree of unpredictability of generated paintings. As I mentioned earlier, it can be achieved in two ways:

By changing the method of random value generation

By changing the rules for filling in the spaces based on a generated number.

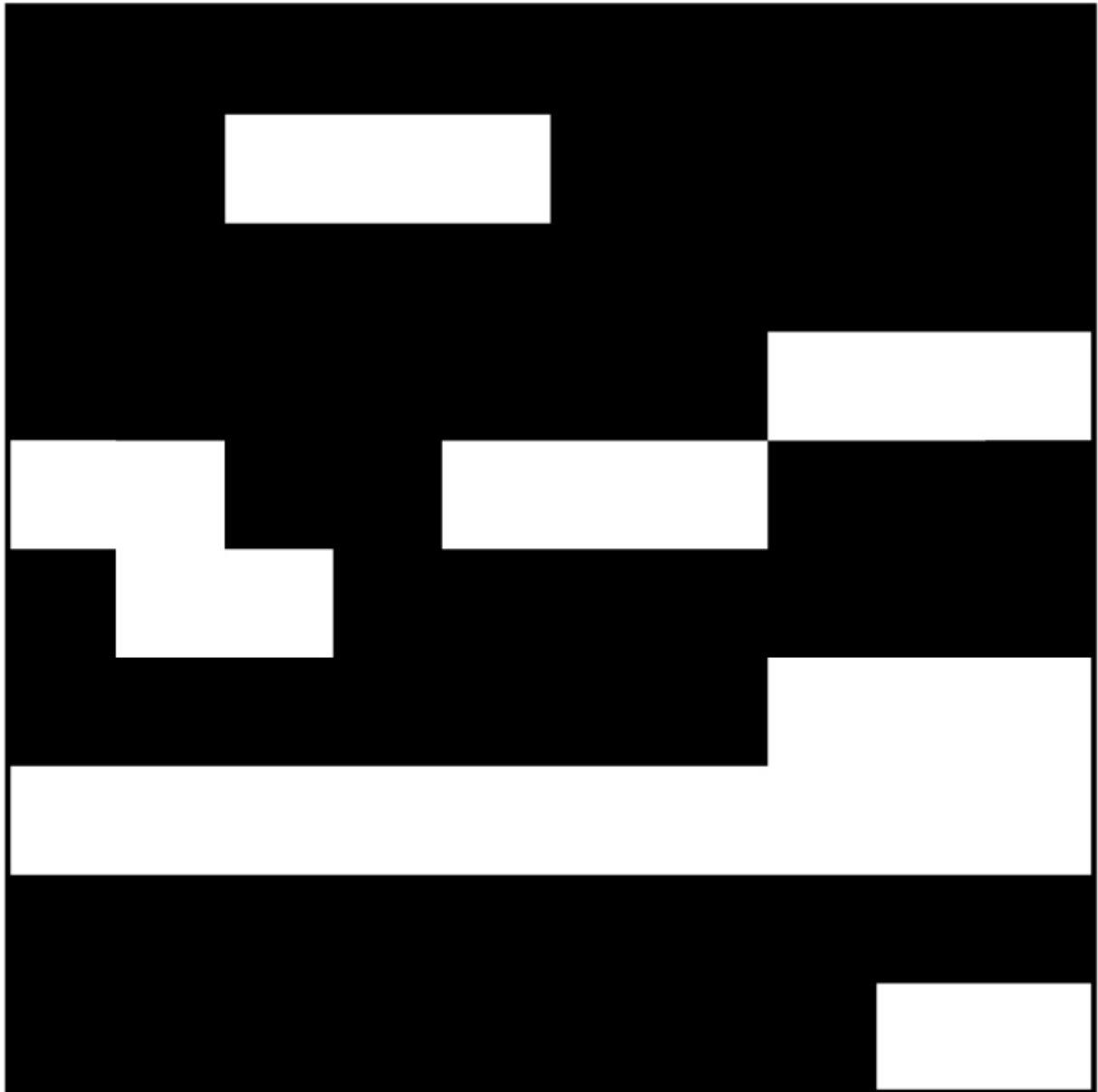


A new randomized algorithm:

A piece of white canvas has been divided into 100 equal squares to form a 10 by 10 grid.

1. The artist rolls a six-sided dice.
2. If the result is even, the number of squares equal to the dice roll is painted black.
3. If the result is odd, the number of squares equal to the dice roll remains white.
4. The artist chooses the next square.
5. Return to step 1.

(dice rolls: 4,6,2,3,4,4,4,2,6,2,5,2,3,4,1,1,4,6,2,2,3,5,5,4,2,6,2,4,3)



As a consequence, a structural complexity has been achieved, which was impossible to do before, revealing a completely different graphic character of the painting with a distinct predominance of black due to higher even numerical values. Individual paintings created by means of the algorithm will not be identical. Despite being similar in character, their final shape will always be relatively unpredictable. The above examples illustrate that both the randomization mechanism and the decision-making algorithm forming a work of art significantly affect the end result. Nevertheless, regardless of the randomisation method, the algorithm has a decisive influence on the graphic form of the work, creating a set of rules and restrictions as well as allowing a determined range of randomness.

Analysing the computer process of generating pseudorandom numbers, it is easy to discover its flaws concerning the degree of unpredictability. There are also several examples of randomization methods found in a real world such as the aforementioned coin flip or six-sided dice roll. On the one hand, these processes are problematic to use in cooperation with a computer but, on the other hand, the actual random element here is based on the human and, in a sense, material factor. The randomness of a dice roll results from a variety of determinants such as the force and height of the throw or material resilience. In the ideal environment where all the factors can be controlled, it would be extremely likely to produce a high repeatability of the

result. The complexity of the randomization conditions affects the degree of predictability of the generated value. Therefore, the question arises as to how to achieve a comparable degree of unpredictability utilizing digital media. While attempting to find a solution to this issue and examining the applications of various randomization mechanisms, I encountered the concept of noise.

In the world of digital technologies and electronic systems, noise is a common, generally undesirable, phenomenon introducing unwanted values. Whenever it appears in audio technologies, which use a sound signal or amplify sound, it is followed by persistent attempts to eliminate it. In the recent era of analogue television, image noise was omnipresent, especially visible in shades of grey on a channel which was not tuned to the corresponding frequency. Noise can also be heard during a radio broadcast or in any form of a wireless audio transmission. The movement of electrons in an electromagnetic wave is highly unpredictable on condition that it is not formed into a proper amplitude. In addition, interference can be caused by any electronic device emitting its own electromagnetic radiation as well as frequencies containing the cosmic radiation noise, which is largely the remnant of the Big Bang. All the chaotic data can be utilized as a source of a random signal with an incomparably high degree of unpredictability. These data are simply unpredictable power surges which can be easily converted into numerical data of any range by means of computer. Another source of noise is an electron movement in electronic semiconductor components called 'shot noise'. The effect, though generally unwanted, can be deliberately produced, thereby obtaining the so-called white noise generator (often used by specialists to test the range transmitted by speakers in sound systems). The availability of this technology as well as possibility of customizing its diagram to meet my needs were the deciding factor in selecting the idea of using the unpredictable movement of electrons to produce a work of art of a procedural and random nature. An additional factor determining the use of the technology in question is its unrepeatability, which makes it superior to the application of computer-generated pseudorandom numbers or noise emitted by a radio receiver. As I mentioned before, two different computers simultaneously generating a pseudorandom number always produce the same value. On the other hand, radio noise at a specific time and place is an outside signal delivered to the receiving device which is not a generator but a receiver of random variables. As was the case with two computers, two radio receivers in the same place and time receive exactly the same transmission; however, two noise generators manufactured from identical factory-made components, regardless of space and time, never generate the same values. From the conceptual point of view, I am satisfied with the fact that I am responsible for building the device which initiates the process of generating random variables.

It is generally believed that distinguishing a work of art in which the numbers have been generated by a roll of dice from the one in which the numbers have been generated by a computer or pseudorandom algorithm is virtually impossible. Computer algorithms are specifically designed to follow the rules of probability theory and closely simulate the environments of random events. Although it is true from the mathematical point of view, a work of art is characterised by the concept and ideological sphere. Therefore, in my belief, there is a major difference between using noise which illustrates the performance of an mathematical algorithm and computer's clock and the observation of noise in the form of a graphic image depicting a chaotic electron movement.

Contemplating a number of examples of both types of noise depicted in the form of graphic variations of my works, in which the projections are filled with noise in the shades of grey, I clearly discern a significant difference in their characteristics. In my belief, the noise initiated by the electron movement is not only incomparably more beautiful but also it is characterised by a unique indescribable organic form although I am fully aware of the fact that an average viewer will not appreciate it as I do.

ALGORITHM

Algorithms accompany people's daily routine in a number of ways. It happens so often that one does not realize that they are used in such mundane tasks as cooking a soup according to a recipe which means following instructions to achieve a specific goal. Algorithms are present in activities such as playing games, browsing internet websites, or shopping online; whenever one deals with computer or mobile technologies. It is possible owing to IT specialists and hobbyists who write algorithms in the form of computer code. The above examples address only a practical sphere of algorithm implementation. However, a number of algorithm designers find beauty in the algorithms themselves and therefore their performance or results become irrelevant. The same concept relates to art although artists too use algorithms consciously and unconsciously.

An algorithm may play the role determined by mathematical definitions, however, in some cases, its notion is not easy to comprehend and the process of its performance and execution of instructions does not have a clearly defined ending. In textbooks on mathematics and IT, the definitions of an algorithm seem to limit this term to the role it plays in the corresponding disciplines. In general, an algorithm is required to complete its task in the least complicated possible way. Therefore, it needs to be concise, have a determined number of results, and, above all, end its performance at a defined moment.

In the absence of a more exhaustive definition encompassing all its potential applications, I will attempt to develop my own broad definition of an algorithm in order to apply it in the fields of art and all kinds of hybrid activities. My aim is to categorise algorithm applications in artistic activities so as to clearly present how the performance of an algorithm is manifested in my own works, especially the ones included herein.

Definition and its explanation

An algorithm – a set of instructions to perform a logically defined task, with previously determined, unchanging elements.

The definition thus formulated allows for a great deal of freedom to classify activities within its scope. Despite the concise form, one is able to identify the key elements of an algorithm to perform properly. Firstly, an algorithm must be a set of instructions which can consist of any number of elements, including a one-element set with one instruction.

Secondly, an algorithm must be delivered to an object which performs an operation although I do not define the form of transmission, leaving the matter open. It is obvious that an algorithm ought to have comprehensible representation, however, apart from notation, I allow here various forms of transmission such as verbal, which is conveyed directly or, for example, via a recording as well as any system of sign or signal codes.

Thirdly, it is crucial to point to the immutability of the transmission form. It may be perceived as intuitive, however, I need to eliminate situations in which instructions are transmitted in a variable way during the algorithm performance. The form of algorithm transmission should be determined prior to its notation or transmission and must be comprehensible to the executing object. Before writing an algorithm in the form of code, a programmer must be familiar with a specific programming language and its rules in order to 'pass on this knowledge' to a machine. A similar rule applies to a chef who wants to follow a recipe to prepare a specific dish.

I intentionally do not impose the requirements concerning the number of end results as I am aware of the situation in which the algorithm executes a command to return to the previous instructions, thereby creating an endless loop. As in the example of the instructions for producing a random image from the chapter Randomness, it is clear that the algorithm designed in this way, not limited by the size of canvas and the number of squares, can theoretically perform endlessly without producing the finite number of results.

The further discussion requires presenting a simple division concerning the form of algorithms in an artistic work in terms of their application. The division does not exhaust the issue of the algorithm application in a piece of art because of situations in which artistic activity utilizes algorithms in each of the presented functions or combinations of them. It is also important to emphasise the fact that the following division does not attempt to illustrate or recreate the existing concept of 'algorithm art' or to classify the existing works within its field. My aim is only to identify and categorise the applications of an algorithm in artistic activities in general.

Algorithm as a tool

The use of algorithm is often unconscious, and despite the knowledge about the existence of the term 'algorithm' or the ubiquity of its applications in IT technologies, computer software is used without considering how it operates and what mechanisms control it. Technology is required to perform a specific task and its operating principles are only known to a limited number of people. Algorithms are regarded as ready-made solutions created as a tool by third parties and nobody raises a question of what exactly they are. In fact, graphics software, not to mention Internet technology, is a highly complicated algorithm. The category in question includes all the activities which utilize computer software in order to achieve an artistic, graphic or interactive effect. It is, therefore, important to separate the sphere of a work of art, its graphic, formal, or conceptual form from the algorithm performance. Here an algorithm is neither a piece of art nor the principle of its performance. It is a tool in the hands of an artist, like a chisel in the hands of a sculptor which was made by a craftsman and does not possess any conceptual sphere enriching the work.

Algorithm as an instruction

In this case, an algorithm falls outside the scope of IT definitions and acts as a manual for an artist or artist's instructions for a viewer. It is a representation of the conceptual sphere of the work which can transform into all kinds of objects or artistic activities. Once again, the chapter Randomness provides an example of a painting produced by means of random dice rolls or coin flips according to specific rules, in other words, an algorithm. Surely, it is a well-known fact that Ryszard Winiarski, while creating his works on the basis of random mechanisms, acted according to specific instructions, which, written on a piece of paper, placed on the other side of the painting produced in accordance with a particular algorithm. In such cases, the algorithm can be executed iteratively, however, regardless of whether it produces similar or significantly different graphic effects, it is the algorithm that plays a decisive role in the creation of a piece of art. It is a kind of matrix, die, or formula defining the limits of variations of its form. On the other hand, the category includes purely conceptual activities devoid of effects in the real world, perhaps, except for the representation of the algorithm itself or such examples that will result in ephemeral and short-term actions. A flash mob, which is initiated by the author's instructions for the participants, is an example of such an artistic phenomena. There are also examples of conceptual works in the form of an algorithm – instructions directed to a viewer, not always requiring their implementation, but still based on ideological premises. The same applies to the concept of event scores, which dates back to the beginnings of conceptualism; they are scenarios of performative events, consisting of simple instructions devoid of elements characteristic of a script, such as dialogues.

La Monte Young's *Compositions 1960* is a set of instructions resembling the formula of an algorithm. Although the instructions were often impossible to follow by a viewer, it did not deter, for example, Nam June Paik from interpreting one of them ('Draw a straight line and follow it') using a chamber pot with ink and a tie on his own neck. In addition, the book *Grapefruit* can be cited as another example of this type of artwork, comprising simple instructions which can (but do not have to) be followed – a brilliant conceptual idea by Yoko Ono.

TUNAFISH SANDWICH PIECE

Imagine one thousand suns in the sky at the same time.

Let them shine for one hour.

Then, let them gradually melt into the sky.

Make one tunafish sandwich and eat.

- 1964 Spring

In conclusion, the category includes the works in which an algorithm is the medium for conceptual and ideological sphere of the artwork and the effect of its performance may, but does not have to, exist in reality. An algorithm is present in the form of precise instructions, and the nature of its notation does not affect its reception.

Algorithm as a work of art

In this case, the algorithm becomes an artistic object; not its instructions, not its conceptual sphere, not even the effect of its performance, but the form of the algorithm itself. The algorithm performance may result in displaying a text on the computer screen, performing a simple computation or complex graphics operations and it is the form of its representation that will be exhibited as a work of art. It can be illustrated by the examples of artists who apply computer code in order to create interactive installations, procedural animations or a variety of graphic forms using programming languages, and, what is the most important, their works are inseparably accompanied by a printout or a poster depicting the code itself. Computer code has its graphic and typographic characteristic, a repeatable structure which can be considered exceptionally aesthetic. In addition, code as a text layout can be freely transformed as the so-called whitespace characters are completely insignificant. Therefore, depriving code of all the spaces, tabs and newline characters will not affect its performance. In the field of a formal computer algorithm, the freedom to format code allows for individual modifications in the representation of the code itself. Experienced programmers, for example, tend to format code to meet their needs so that it becomes their hallmark.

Since such modifications are possible, it is obvious that artists who use programming languages as a medium will appreciate the graphic representation of an algorithm, which embodies their artistic ideas. The representation is neither the instructions for an artist nor viewers. In this instance, computer code performs or has performed its task as a tool or mechanism controlling the work of art, and its printout becomes an artistic artefact including, but not displaying, the conceptual sphere of the work. The vast majority of viewers who are not familiar with IT issues will appreciate only the aesthetic value of the algorithm and find its form mysterious and intriguing.

In his artistic undertaking Computer Virus Project, Joseph Nechvatal introduced a similar concept. In a computer simulation, the computer algorithm acts as a virus corrupting the graphic form of the digital reproductions of the artists' earlier works. In addition to the animation, the exhibition showed the algorithm which controlled the virus in the form of computer code. It was displayed on a huge, over 20 metre wide canvas in a landscape orientation, read from top to bottom, not from left to right, which additionally decreased its legibility value and transformed it into a graphic and typographic artefact.

For a number of artists using a computer algorithm in the form of programming languages, the code itself becomes an object of admiration. The code, not the result of its performance, is the essence of an artwork. Furthermore, there is a field of computer science in which a computer algorithm has freed itself from the limitations of the definition requiring simplicity and programming languages are developed so as not to be user-friendly. The designers aim at creating programming languages which are unconventional, difficult to comprehend, mocking,

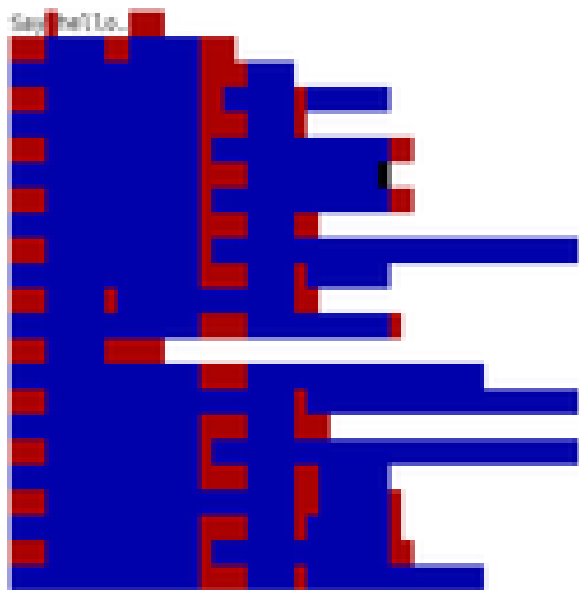
and, above all, unique. Esoteric programming languages created their own niche in the field of the programming world. I believe the sphere, which exists on the borderline between art and computer science, still has not been sufficiently studied as an artistic conceptual value. Computer code allows a programmer to execute computer commands. Therefore, its linguistic form serves only one purpose: to be comprehensible to a human. Computer code written by a human is translated into machine code, comprehensible to a computer, by computer software called a compiler. Code can take a text, digital, or graphic form on condition that it has been written logically by a person, assuming that it can be translated into a machine language by a compiler.

The following examples illustrate a computer algorithm in various forms of esoteric programming languages. The result of the algorithm performance is not significant (generally, it is code generating a common IT phrase 'Hello World' on the computer screen). I believe that the form of the algorithm representation is both a graphic and conceptual work of art.

Brainfuck

```
+++++++>++++[>++++>++++>++++>+<<<<-]
>+>+>->>+<]<->>.>---.+++++++..+++.>>.
<-.<..+++..-----..-----..>>+.>+.
```

Whitespace



Piet



All the above examples show real programming languages in which the code takes the form of symbols, characters, or colour values. In the computer world, they are simply numerical values which are capable of transmitting any information.

FORMULA

The concept of a formula may seem to be similar to an algorithm, however, it would be inappropriate to unify their meaning because, whereas an algorithm is a set of instructions, a formula, according to the principles of mathematics, physics, or chemistry, is a representation of a phenomenon manifested by symbols. A formula explains a phenomenon invariably; for example, the area of a square formula defines any existing square in any form and physics formulas describe immutable phenomena and properties of the material world. A formula may initiate the process of creating an algorithm which confirms its performance, however it is not an instruction, only a symbolic representation. An algorithm, on the other hand, is its outcome. I will illustrate the above with one of the most well-known formulas for subsequent numbers in the Fibonacci sequence, the principle of which is that each number is the sum of two preceding ones. The graphic representation of this sequence is a characteristic spiral defining the golden ratio of the plane. Omitting the numbers 0 and 1, for which the values of this sequence are successively 0 and 1, each consecutive number of the sequence can be calculated according to the formula:

$$F_n = F_{n-1} + F_{n-2}$$

Replacing 'n' with consecutive numbers, from 2 to any number, results in an invariable set of numbers according to the formula. The beginning of the sequence is thus:

0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181...

Now, it is time to investigate an algorithm containing a list of steps to produce the identical result:

- Take a sequence of digits 0,1
- Sum up the last two digits of the sequence
- Place the result of the addition at the end of the sequence
- Return to step 2

The examples perfectly illustrate the differences between a formula and an algorithm. What is more, they characterize an algorithm as a form of translation of a formula into a set of instructions. These instructions can be easily translated into computer code. Below I present an example of a program in Processing language which, in the computer console, produces 10 consecutive numbers of the Fibonacci sequence:

```
int F0 = 0;
int F1 = 1;
int Fn;
int liczbaWyrzow = 10;
// a loop repeated a number of times equal to a variable 'liczbaWyrzow', see below
for( int i = 1; i <= liczbaWyrzow; i++)
{
  println( F1 ); // print line - command for printing a line of text
  Fn = F0 + F1;
  F0 = F1;
  F1 = Fn;
}
```

According to a similar scheme, in my own works I use and convert, to satisfy my own needs, existing mathematical formulas by means of computer algorithms to generate specific geometric patterns: circles, lines, Bezier curves of a procedural nature. Consequently, the next chapter offers a comprehensive explanation of procedure and procedurality.

PROCEDURE

Another term which is semantically similar to an algorithm is procedure understood as a defined method of operation. Contrary to an algorithm, procedure describes a process of conduct in specific situations, although, obviously, it can be represented as an algorithm. Therefore, procedure can be an algorithm or its applied element in a specific case.

Recently, in the world of computer graphics, the term procedurality has begun to form and disseminate. The expressions procedural graphics and animation are gradually replacing terms such as real-time or generative animations. What is important, all of these terms share a common idea which I would like to elaborate on in this chapter.

The introduction of programming in the field of computer graphics displayed by means of a screen has created unprecedented opportunities for IT specialist, hobbyists, computer game developers as well as artist, especially in the field of animation. Traditional animation has always required a huge amount of work. Once determined, the animation sequence was immutable in its form and left no space for improvisation. Its creator and viewer know exactly what it would look like next time they see it.

The introduction of programming in the field of animation has enabled the use of pseudorandom algorithms in the description of the properties of animated objects. Animated object no longer have to follow the same pattern, gaining almost autonomous life. Often the authors themselves cannot predict their movement (it is sufficient to include an element of arbitrariness in the control algorithm). Each frame of the programmed animation can be almost completely unpredictable, unique and 'random', and this is the main feature which distinguishes it from traditional animation as well as video editing. A new term has to be coined to separate these two phenomena. One of the first was the term real-time animation. Unlike the traditional techniques, the decision concerning the contents of the next frame is made at the moment of its viewing.

In addition, the issue of graphics or procedural animation includes the complex process of real-time image generation. I would like to illustrate it with an example drawn from the commercial world although I consider this idea to be an excellent example of conceptual art.

In 2016, the English Hello Games studio launched a new product under the name No Man's Sky on the gaming market, revolutionizing the method of creating the world explored by a player. High-budget games are usually produced by studios employing dozens or even hundreds of computer graphic designers who model even the smallest elements of the game world with meticulous care. No Man's Sky is a 3D science fiction game in which the player explores the universe, visits planets, collects the necessary materials for base construction, and improves the character's equipment. It has been calculated that visiting each of the available planets for at least one second would take the player five billion years of real time. Furthermore, each of these planets is exceptional; it has a unique flora, fauna, and geology. Being supported by an army of graphic designers, the creation of similar content would be unfeasible to accomplish in 100 years.

Therefore, the question arises as to how a three-person studio did the impossible. Unsurprisingly, it was achievable owing to algorithms, formulas, and procedures. A brilliant algorithm was applied to generate a virtual universe, in which all the elements including interplanetary space, space stations, asteroids, and finally planets with their environments, on the basis of simple numerical input data (seeds). The X, Y, Z coordinates (the position of the player in the

3D reality of the game) were used as seeds, which also allows to maintain the cohesion of the world so that each player within the same coordinates will explore identical content. The random generation is not applied because the position of each element contributing to the game universe is predetermined by the algorithm. On the other hand, one can experience a huge amount of effect unpredictability. The creators themselves did not see even a per mille of the universe they had created, and the generating algorithm is so complicated that its designers are not able to predict what can be encountered within a short distance from the universe centre.

Therefore, procedurality is a term which perfectly describes phenomena in the field of computer graphics, including all forms of pseudorandom generation of real time and containing three basic components:

- Seed – input data (generally alphanumeric)
- Control algorithm – a set of instructions which process input data into the end result (mainly in the form of computer code)
- The end result (most frequently in the form of graphic computer animation)

In addition, it can be specified that in the case of procedurality, once the control algorithm is determined, it processes input data in the same way. Similarly, looking back at the first chapter and considering the example of the first algorithm generating pseudorandom numbers based on any seed, one can observe that it has a procedural character and the generated numbers can be referred to as ‘the procedurally generated numbers with a certain degree of unpredictability’.

I believe it is a sufficient reason to depart from unequivocal classification of procedural works in terms of randomness. What is more, I perceive randomness as an idea that can never be fully achieved. In my view, true randomness is a reflection of a degree of unpredictability, which would create a realistic chance, if it existed in the real world, to start a combustion engine, with repeated attempts, after filling up the tank with water instead of petrol.

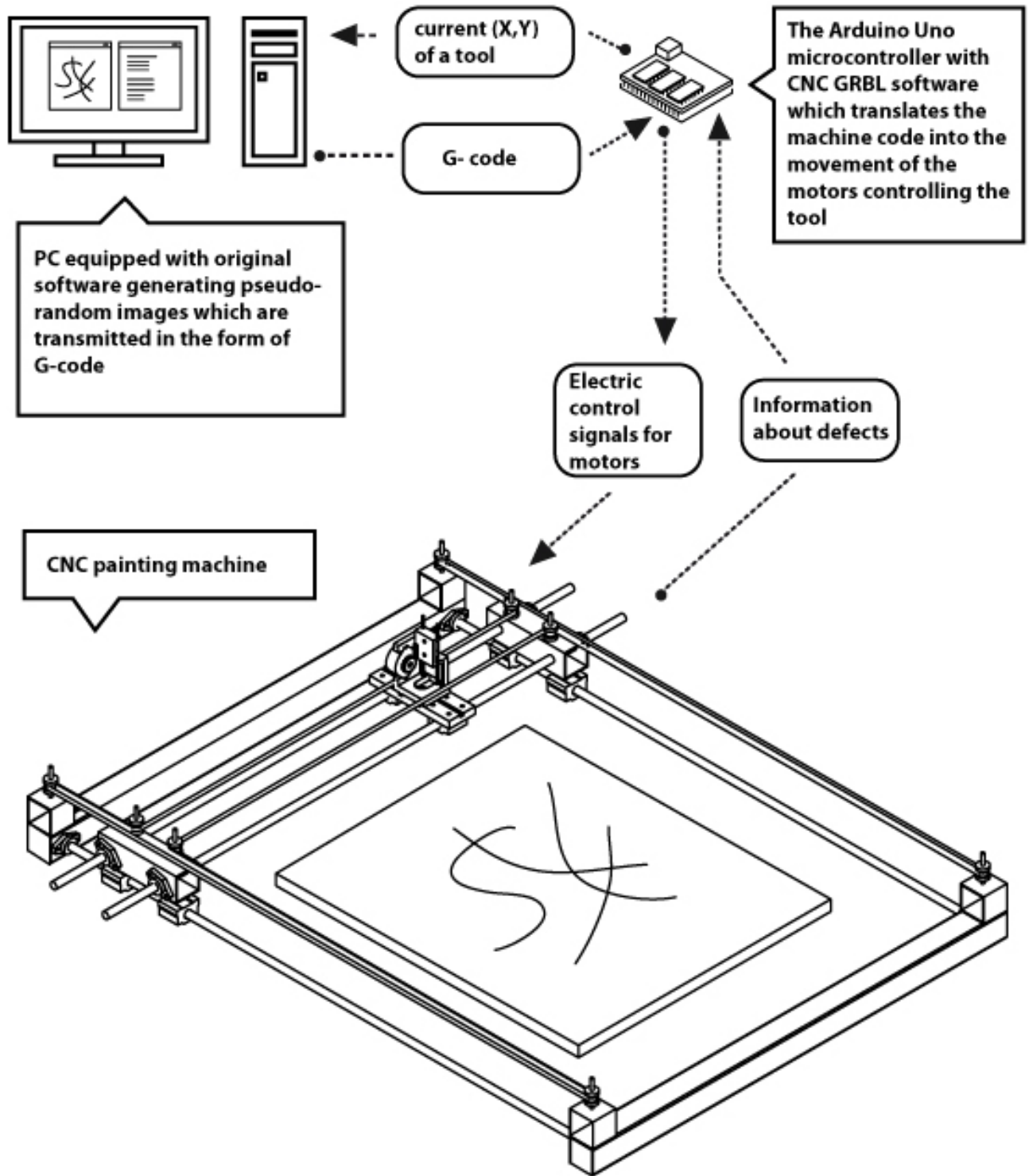
DESCRIPTION OF WORKS AND APPLIED TECHNOLOGIES

In order to be able to study pseudorandom processes with a high degree of unpredictability in a large number of cases of painting compositions, I was forced to eliminate the human (artist) participation as a possible additional variable in the algorithm performance. After dividing the process of producing a painting by means of traditional techniques into simple stages, one is able to observe that the artist's hand moves in three dimensions of the working area – the X and Y axes of the support plane and the Z axis, controlling the pressure of hand or taking the next portion of colour. In order to eliminate artist's participation in the painting process, I had to create mechanical environment allowing a painting tool to move in a similar way in the X, Y, Z axes using the simplest technological means.

The solution which meets the above requirements was the use of a CNC machine – a technology popularized on a large scale in recent years by the community of constructors, hobbyists as well as specialist suppliers of machines for 3D printing and cutting. In simple terms, a CNC machine is based on three transmission axes (for each of the X, Y, Z axes) allowing the head with a tool to move to any point on the X, Y axes as well as changing its position on the Z axis by lifting and lowering a tool (in my case, it is a traditional medium in the form of a brush applying acrylic paint).

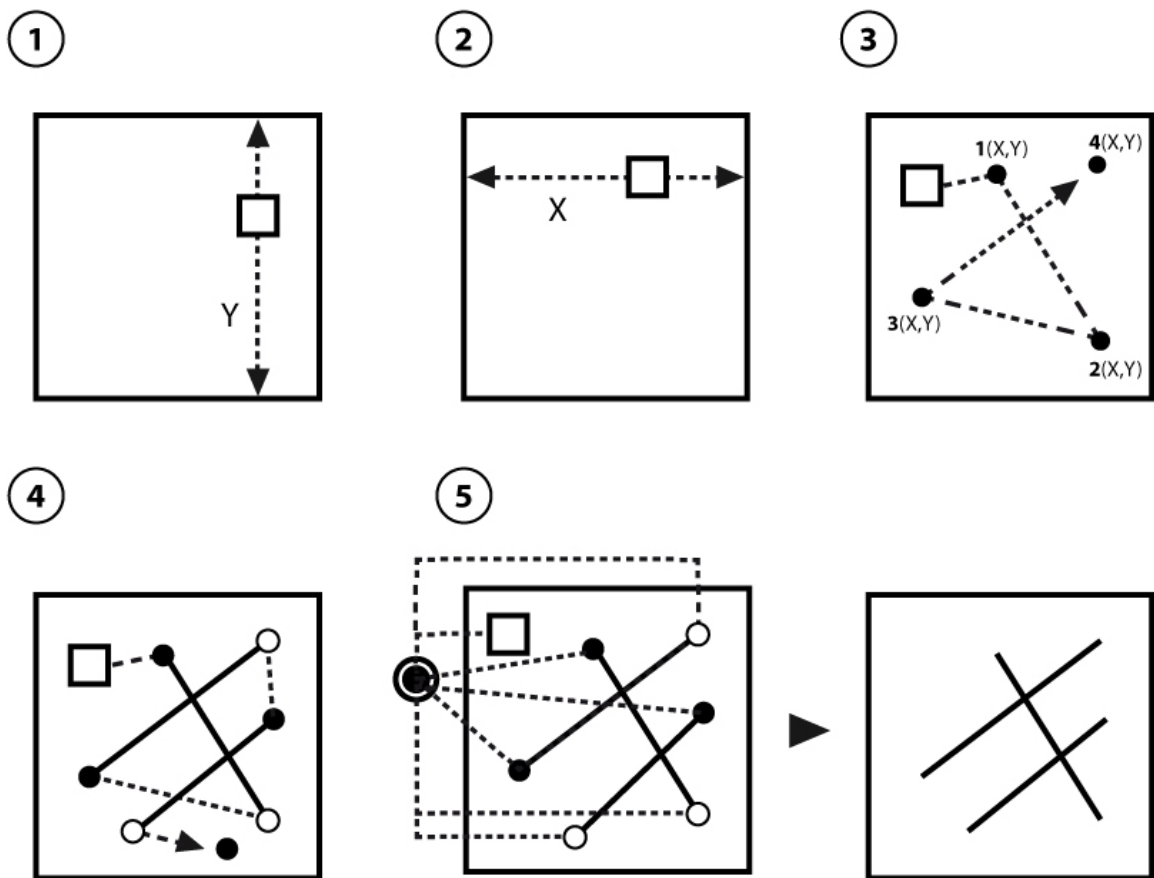
In CNC machines, the motion of the head is controlled by the software installed on a computer which sends the consecutive X Y Z coordinates taken from the project file prepared beforehand using graphic design software. If one wishes to paint a circle, it is sufficient to prepare a drawing, for example using a computer program Illustrator (1:1 scale) and import the file into the machine software. The machine tool, which is set in the rest position in a corner, is raised on the Z axis and moves into the starting position without touching the canvas. Then, the tool is lowered and the software sends consecutive X Y coordinates corresponding to the consecutive points which outline a circle on the surface. Having drawn the circle, the tool returns to the rest position or moves to the paint container to take a portion of paint and continues its work on condition that another task has been set. The control software for this type of machines is widely available and even free of charge owing to open-source licences and, as in the example mentioned above, it allows to transfer compositions from previously prepared files. In the case of my work, it was necessary to design real-time decision-making software whose aim is to determine the next step executed by the machine.

The diagram below shows the ideological form of the operation of the machine, which I designed, together with a computer and software:



Having constructed the physical structure of the machine, the first step was to enable the movement of a drawing or painting tool within the working area (canvas) using basic software in the form of a collection of machine instructions instead of conceptual algorithm. Here are the basic steps the software has to execute, assuming that the working area is a plane with X and Y axes:

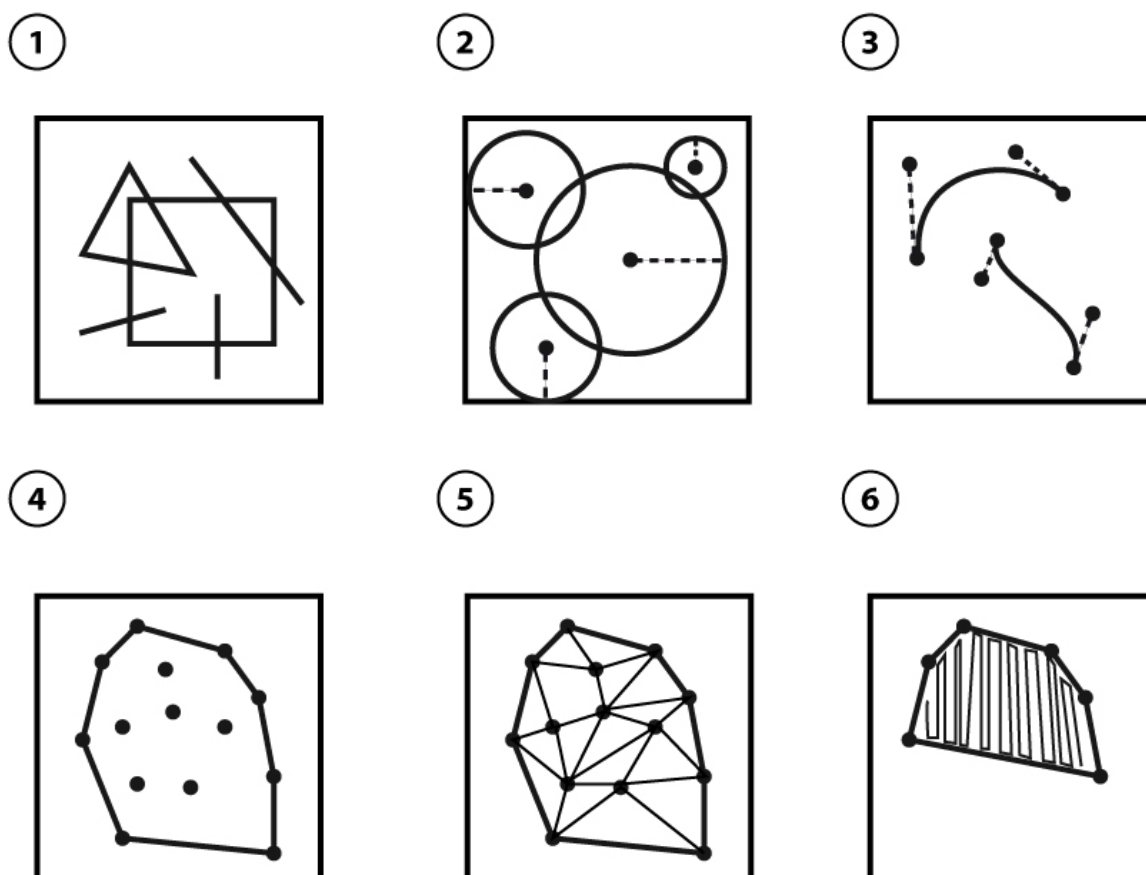
1. movement along the Y axis
2. movement along the X axis
3. movement along both axes simultaneously following consecutive points (X, Y)
4. lowering and lifting the tool – movement along the Z axis, distinguishing between moving to the beginning of a new line and lowering the brush on the canvas to paint
5. in the case of a tool in the form of a brush, it is necessary to anticipate the need to move the tool to the paint container and then return to painting



Having prepared and tested the functionality of the basic control program, I commenced developing an algorithm capable of generating shapes and geometric patterns delivered to the machine as subsequent points-coordinates on Y X coordinate system.

Prior to converting into computer code, I developed an ideological algorithm allowing a software-machine team to generate both basic and complex geometric figures. Applying various mathematical formulas, I created algorithms generating geometric shapes with variable properties, which could be produced on the basis of pseudorandom variables in the further process. Having enabled the software to generate specific shapes and geometric patterns, the only step to take was to design a randomized algorithm determining which of the available actions would be used and how many times. In order to create more com-

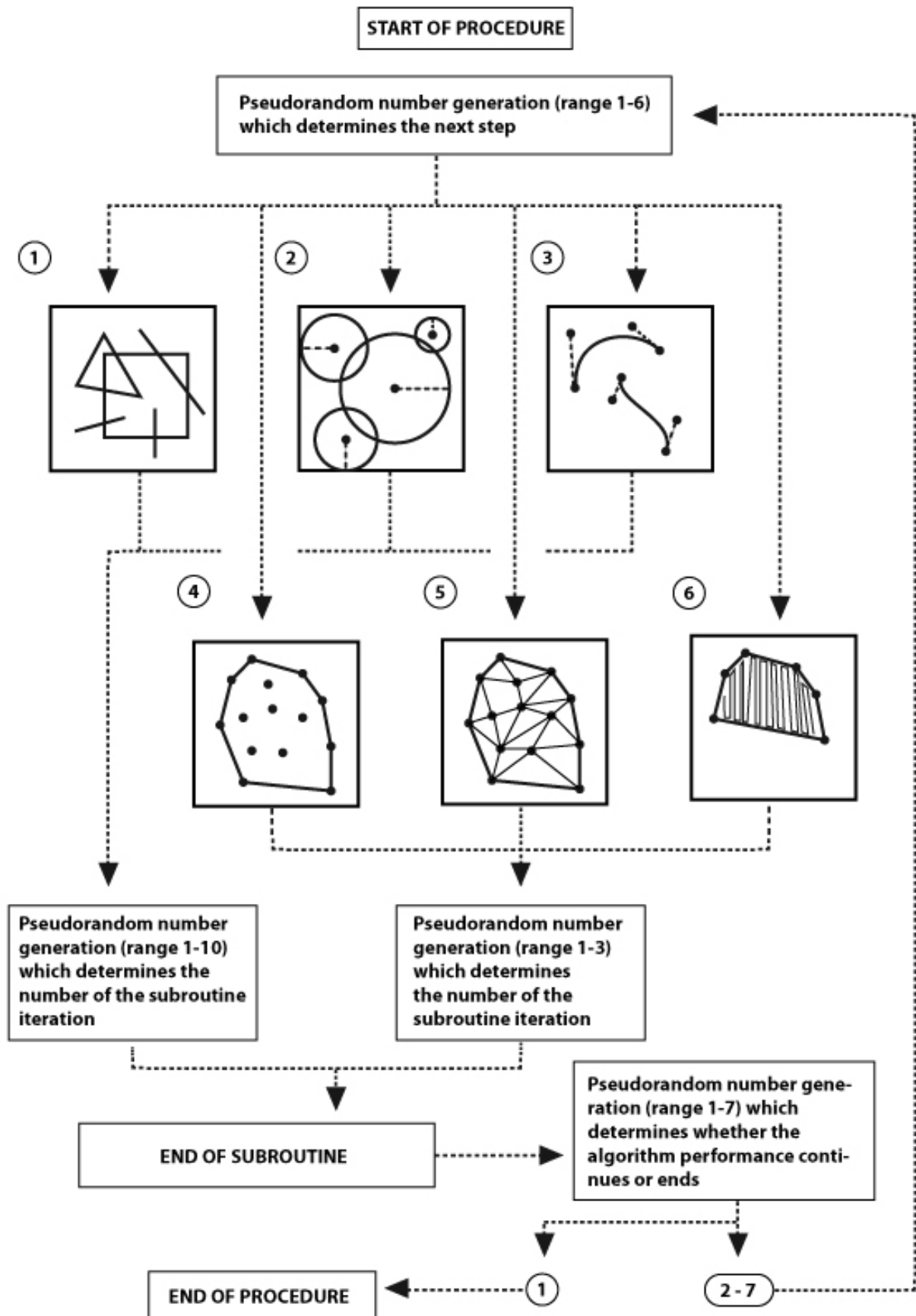
plex patterns, for instance organic forms, I had to use ready-made mathematical formulas in computational geometry and convert them into the form of a computer algorithm controlling the machine. After equipping the machine with a tool enabling it to move from point A to point B, it was possible to generate simple geometric figures (1). However, in order to produce ellipses and circles, I had to use an orbit equation (2). Another important tool was the randomly generated Bezier curves. Their application allowed to control curvature lines (3). I also applied formulas for finding the convex hull, which allowed me to generate more complex and random shapes (4) as well as the Delaunay triangulation – searching for the division of a set of points to create a triangular grid (5). Finally, the last subroutine I selected was to develop an algorithm which was able to fill the given shape with parallel lines creating fairly regular patches of colour (6).



The process of equipping the software with appropriate drawing algorithms resembles the process of teaching a child how to draw a variety of geometric figures. The next step is to raise child's awareness of the decision-making process – what is going to be painted next – based on pseudorandomness. In addition to a large number of subroutines and endless possibilities to combine them, high unpredictability of a painting tool in the form of a brush and paint created a high degree of unpredictability in the composition of the painting. For this reason, I decided to temporarily abandon the idea to complicate the process of generating random data as well as using a noise generator in favour of pseudo-random numbers generated by computer. In the course of experimenting with various forms of algorithms, I selected two methods of work with the composition and composition decisions of an algorithm:

- Pseudorandom algorithms
- Sequential algorithms

The diagram below illustrates the decision-making processes of the final form of the pseudorandom control algorithm.



Below I will explain the performance of the sequential algorithm, which has become an important element my work. I discovered it accidentally in the course of an experiment and although it does not play a crucial role in my work, I believe it has enormous potential.

In the course of my study on a random deformation of a geometrical shape such as the process of deforming a circle by noise algorithms (Ken Perlin's Perlin noise, in particular), I discovered the possibility of sequential transformation of a geometric form. The algorithm performs in a similar way to the pseudorandom algorithm described in the chapter Randomness. Using the seed provided by a user, it generates pseudorandom output data and, of course, providing the same seed results in the same effect. Processing the points (their coordinates) constituting a circle or a polygon by a similar algorithm with a given seed results in a specific deformation of this shape. Consequently, for number 1 a repeatable deformation of a circle is achieved and using always the same seed produces the same outcome whereas for number 2 the result is completely different. I decided to search for such transformations for as many numbers as possible and make assumption that with an infinite number of transformations (caused by the infinity of seed numbers) one number can define any shape; for example, there may be a number defining the outline of my profile, borders of a country, or the latest car model of any company. In a similar assumption, I discern the sequentiality and seriality of Opałka's paintings as well as the combinatorial universes of graphic possibilities of *Every Icon* by John F Simon Jr.

Apart from tool replacement and decisions regarding the range of colours, the performance of the algorithms discussed above fully determines the form of the painting. Understanding the mechanisms of my painting system allows to comprehend the idea behind the experiment in the form of artistic work.

The selected collection of paintings represents the works produced in the course of the research concerning the machine operation. Not only are there examples of tests on isolated sub-routines such as Bezier curves, straight lines, or Delaunay triangulation, but also the results of complete algorithm performance combining these processes, thereby producing increasingly unpredictable artistic values. Furthermore, testing a graphic effect involved changing the tool from a traditional painting medium in the form of a brush and acrylic paint to various types of markers as well as experimenting with colours of both media.

The ground for all the compositions is invariably a white canvas, size 50 cm x 50 cm which functions as a Petri dish, thereby complying with the laboratory nature of the experiment.

CONCLUSION

By means of algorithms, formulas, procedures, and a whole arsenal of technological solutions, I explore graphic and artistic forms representing the events characterized by a high degree of unpredictability. Every new form of artistic expression is an experiment, study, and exploration of a new aesthetic. I search for new formulas, algorithms, and methods of generating works in order to discover something unthinkable, something beyond my imagination. I believe computer technologies expand the range of traditional media. While painting, sculpting, or using traditional techniques, I was unable to turn off my 'internal software' which is an integral part of my personality. The experiences of life, the limitations of its physical form, instilled syllabus, or biological and physical needs affected every movement of my hand and, as a consequence, I did not discover anything worth exploring while producing a new work. Even the escape from the object aesthetics to conceptual form was the result of the same 'software'. Being unable to break free from my own limitations, I began my quest for a method for separating myself from majority of decision-making processes in producing a piece of art. During the course of my artistic work, I created specific laboratories where I could set graphic boundaries and explore a wealth of events within them. Such an algorithm will never refer to realistic forms.

I find it fascinating how an algorithm generating patterns, which I consider aesthetic, is also capable of producing the ones I would not consider a finished work. It allows me to break free from the familiar world of aesthetics and transcend its limitations in order to see the beauty and richness of the iteration of form.

The created works and algorithms certainly do not show the full potential of technological solutions which were designed and used by me. During the course of the work on their current form, dozens of new ideas and solutions began to form in my mind. Even though I would not have been able to test and apply them all, at the end of this stage of my work, I am convinced that I have finally discovered a medium whose potential can expand the spectrum of my artistic explorations to the extent that I will not be able to predict the final result. At last!

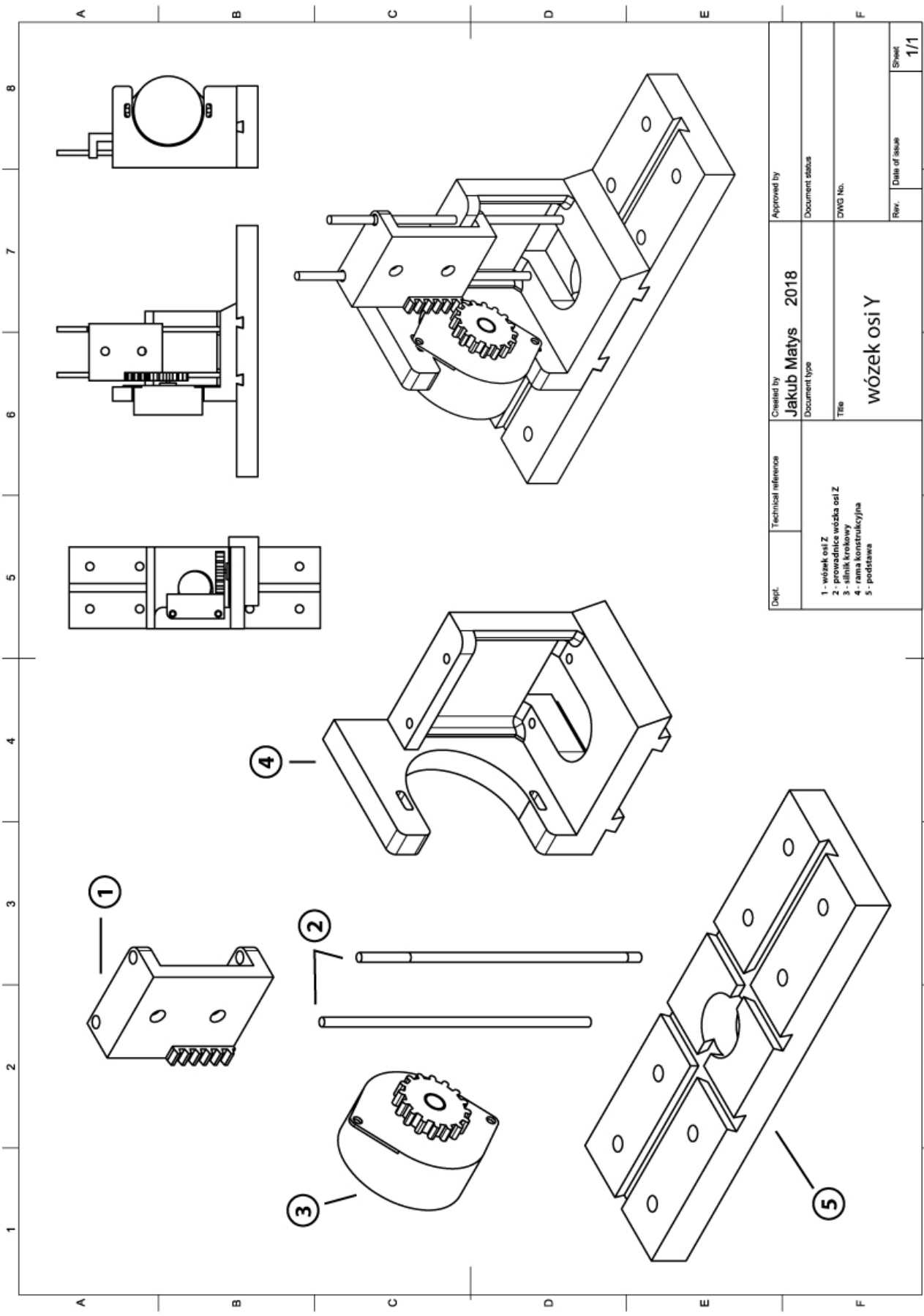
My work as a whole consists of both the created paintings and the process of the conducted experiment. To conclude, I wish to stand in front of an individual piece of art produced by means of my algorithm and technologies and be unable to see my participation apart from the thought that it was I who initiated the artistic process.

THE ABSTRACT OF THE DOCTORAL DISSERTATION

My work is a description of a practical experiment in the field of painting, probability and robotics in terms of a work of art. I describe in it the use of terms and technologies taken from the field of exact sciences as an algorithm, formula, procedure and their applications as an artistic medium. A large part of my work is occupied by considerations about randomness and probability as the main idea of the experiment is to create a system limiting the artist's decision-making in the creation of an artwork using traditional painting media. In this case, computational mechanisms generating pseudorandom events takes over the creative decision-making process. In my inquiries about the probability, I propose to replace the term of randomness, in relation to artistic activities using computer technology, to determine as events of a certain degree of unpredictability, which degree can be characterized by the general gradation like high or low. Another important theoretical element of my work is the description of the algorithm's term and its use in artistic activities and differences in the meaning of the same term in the fields of mathematics or computer technology. I also mention a simple division of the application of the algorithm in art to the Algorithm as a tool, Algorithm as a manual, and Algorithm as a work of art. I also describe the terms of formula and procedurality and their participation in the creation of my doctoral thesis.

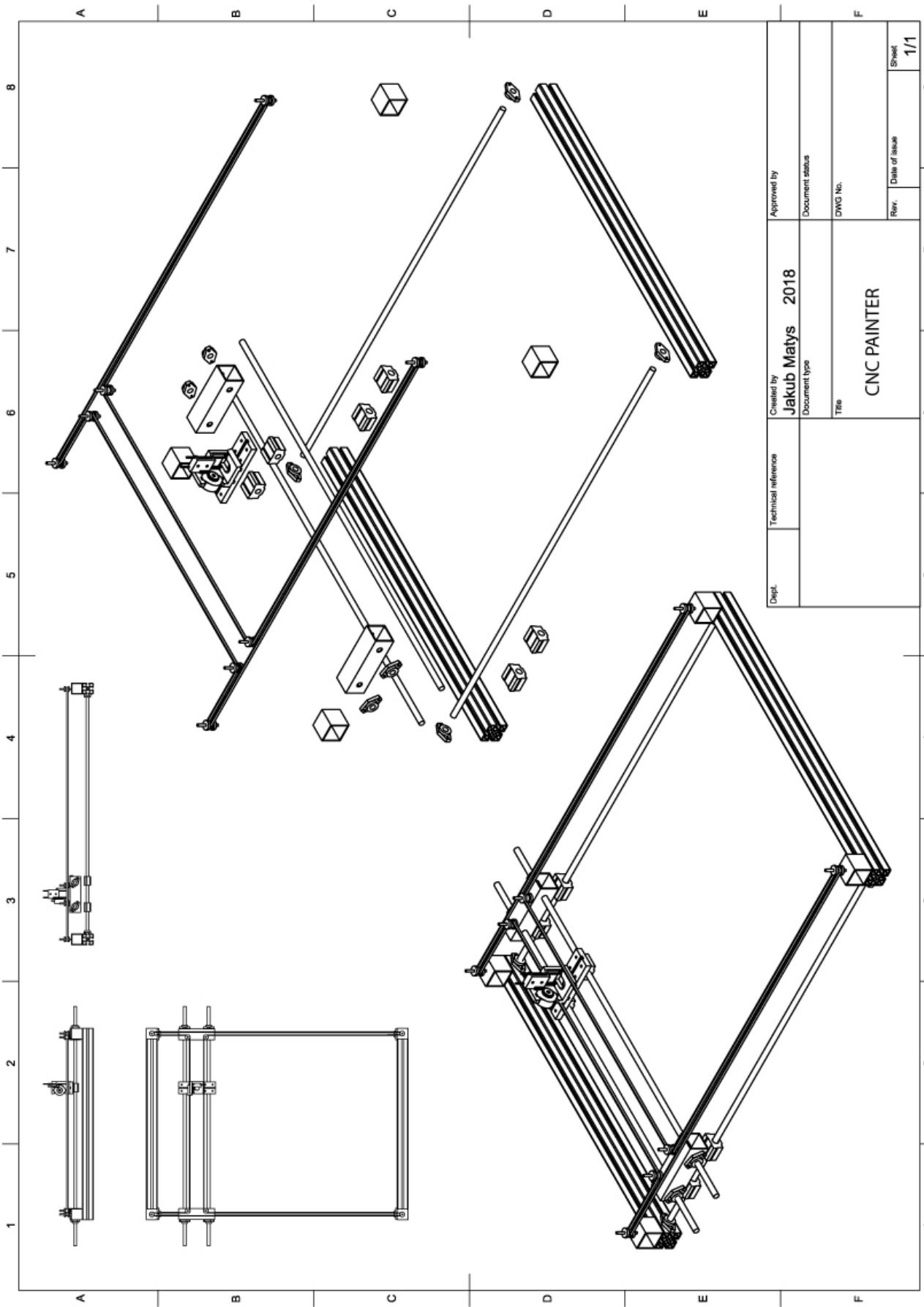
In the practical part, my work includes, built from scratch by me a CNC robot with dimensions of approximately 100 X 70 cm, having the ability to paint on flat surfaces using traditional painting and drawing media, such as a brush and paint or markers of all kinds. The machine is controlled by written by me computer software, generating basic and complex geometric shapes based on mathematical formulas taken from the field of computational geometry. Compositions are created by algorithms based on simple computer-generated pseudo-random numbers. The result of the work of the machine and the software controlling it are 72 paintings with dimensions of 50 X 50 cm that form the painting cycle which is the basis of my doctorate. Observing the operation of the algorithms and the work of the machine and its effect in the form of painting compositions with a high degree of unpredictability, I conclude the conclusion in which, thanks to new technologies as an artistic medium, I see the possibility of standing in front of an individual artwork based on my algorithm and proprietary technologies, free from my participation, maybe besides being aware of throwing the first element of the snowball.

ZAŁĄCZNIKI - PLANY TECHNICZNE



Dept.	Technical reference	Created by Jakub Matys 2018	Approved by
		Document type	Document status
		Title	DWG No.
		Wózek osi Y	Rev.
			Date of issue
			Sheet
			1/1

- 1 - wózek osi Z
- 2 - prowadnica wózka osi Z
- 3 - silnik krokowy
- 4 - rama konstrukcyjna
- 5 - podstawa



Dept.	Technical reference	Created by Jakub Matys 2018	Approved by
		Document type	Document status
		Title CNC PAINTER	DWG No.
		Rev.	Date of issue
			Sheet 1/1

BIBLIOGRAFIA

Monografie:

1. Doris M. Wheatley, Alan W. Unwin, Algorytmy, Państwowe Wydawnictwo Ekonomiczne, Warszawa 1975
2. David Gelemter, Mechaniczne Piękno, Wydawnictwo W.A.B., Warszawa 19997.
3. Mieczysław Porębski, Sztuka a informacja, Wydawnictwo Literackie, Kraków 1968
4. Ryszard W. Kluszczyński, Sztuka interaktywna, Wydawnictwa Akademickie i Profesjonalne, Warszawa 2010
5. Thomas H.Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Wprowadzenie Do Algorytmów, Wydawnictwo Naukowe PWN SA Warszawa 2012
6. Matthew Fuller, BEHIND THE BLIP. ESSAYS ON THE CULTURE OF SOFTWARE
7. Dieter Stempell, RACHUNEK PRAWDOPODOBIENSTWA w ujęciu programowanym, Wydawnictwa Naukowo-Techniczne, Warszawa 1972
8. Rudger Bauman, Grafika komputerowa, Wydawnictwa Komunikacji i Łączności, Warszawa 1989
9. M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, Geometria obliczeniowa Wydawnictwa Naukowo-Techniczne, Warszawa 2007

Słowniki:

1. Lidia Filist, Artur Malina, Alicja Solecka, Matematyka – słownik encyklopedyczny, Wydawnictwo Europa, Wrocław 2007

Katalogi, czasopisma, artykuły:

1. Florian Cramer, CONCEPTS, NOTATIONS, SOFTWARE, ART, 23 marca 2002 Autonomedia, Brooklyn, NY 11211-0568 USA 2003
2. Thomas Dreher Conceptual Art and Software Art: Notations, Algorithms and Codes, grudzień 2005/tłumaczenie ang. marzec 2007
3. Lev Manovich, Cultural Software, lipiec 2011

UNIWERSYTET JANA KOCHANOWSKIEGO W KIELCACH
WYDZIAŁ PEDAGOGICZNY I ARTYSTYCZNY

KIELCE 2018