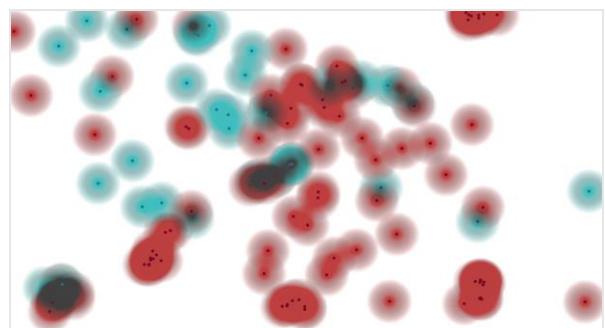
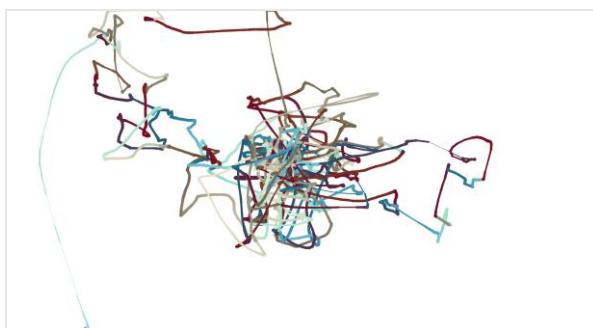
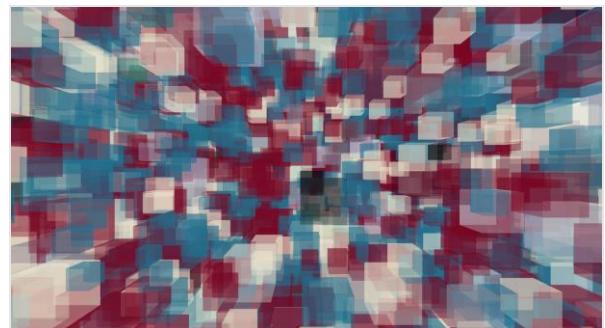
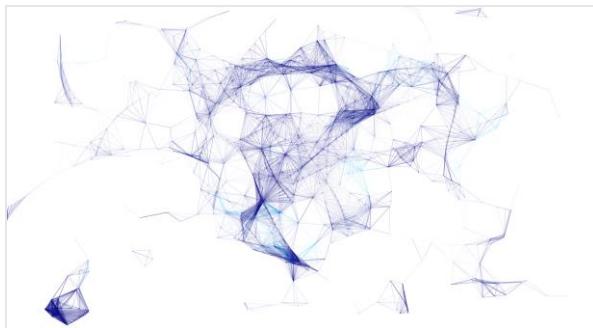


Algorithmische Kunst

durch Maus und Tastatur



Maturitätsarbeit an der Kantonsschule Zürich Nord

Neville Walo, W6I

Betreuerinnen: Hilka Feil, Lotti Bauer

Zürich, Dezember 2016

Inhaltsverzeichnis

1. Vorwort	1
2. Einleitung.....	2
3. Algorithmische Kunst.....	3
3.1 Beispiel für algorithmische Kunst.....	3
3.2 Der Algorithmus.....	5
3.3 Die algorithmische Revolution	5
4. Generative Kunst	7
4.1 Beispiel für generative Kunst.....	7
5. Vorgehen	9
5.1 Grundgerüst	9
5.2 Programme.....	10
5.3 Testpersonen	10
5.4 Testparcours	11
5.5 Ergebnisse	12
6. Analysen	13
6.1 Bildanalysen	13
6.2 Reflexion	17
7. Fazit.....	21
8. Quellenverzeichnis	22
9. Anhang.....	23
9.1 Grundgerüst	23
9.2 Programme.....	25
9.2.1 Programm 1	25
9.2.2 Programm 2	29
9.2.3 Programm 3	32
9.2.4 Programm 4	36

9.2.5 Programm 5	39
9.2.6 Programm 6	69
9.2.7 Programm 7	72
9.2.8 Programm 8	77
9.2.9 Programm 9	81
9.2.10 Programm 10.....	85
9.2.11 Programm 11.....	88
9.2.12 Programm 12.....	91
9.3 Bilder	95
9.3.1 Programm 1	95
9.3.2 Programm 2	99
9.3.3 Programm 3	103
9.3.4 Programm 4	107
9.3.5 Programm 5	111
9.3.6 Programm 6	115
9.3.7 Programm 7	119
9.3.8 Programm 8	123
9.3.9 Programm 9	127
9.3.10 Programm 10.....	131
9.3.11 Programm 11.....	135
9.3.12 Programm 12.....	139
9.4 Von Testpersonen gemalten Bilder	143
9.5 Angaben und Einverständniserklärung.....	146
9.6 Aufsätze	152
9.7 Farbauswahl der Testpersonen.....	158

1. Vorwort

Diese Arbeit steht im Rahmen der Maturarbeit 2016.

Schon lange bevor man sich überhaupt mit der Maturitätsarbeit auseinandersetzen musste, hatte ich alle möglichen Ideen, die ich spannend fand, aufgeschrieben und festgehalten. Zu Beginn war es für mich klar, dass ich meine Arbeit in Sport schreiben werde. Doch je näher der Abgabetermin des Themas kam, desto weiter habe ich mich von dieser Idee entfernt. Ich begann mich immer mehr für die Informatik zu interessieren. Da ich auch schon über einige Programmierkenntnisse verfügte und mir das Programmieren viel Spass bereitet, bot sich diese Richtung besonders an. Zuerst wollte ich mit einem Freund zusammen eine Handy-App entwickeln. Doch diese Idee haben wir schnell verworfen, da wir keine konkrete Idee hatten und es unsere Fähigkeiten überstiegen hätte. Danach versuchte ich mich an zwei weiteren Themen. Einerseits die Kombination von Informatik und Mathematik, andererseits die Kombination von Kunst und Informatik. Bildnerisches Gestalten war eigentlich nie mein Lieblingsfach, denn ich kann weder besonders gut zeichnen noch besonders gut malen. Trotzdem habe ich mich am Schluss für diese Richtung entschieden. Vielleicht auch gerade deswegen. Ich kam mit Frau Bauer in Kontakt, welche mir half, mein Thema einzugrenzen und zu verfeinern. Um auch den Informatikteil abzudecken, bat ich zudem Frau Feil um Hilfe. Sie waren beide sehr hilfsbereit und standen mir mit gutem Rat zur Seite. So hatte ich zum Schluss ein für mich passendes, anspruchsvolles und spannendes Thema für meine Maturitätsarbeit.

Ich möchte mich herzlich bei meinen Betreuerinnen Frau Feil und Frau Bauer bedanken. Mein Dank geht auch an alle Testpersonen, ohne welche ich mein Projekt nicht hätte realisieren können.

2. Einleitung

Ziel meiner Maturitätsarbeit war es Bilder mit Algorithmen zu erschaffen, welche ihre Daten und Variablen aus Tastatur und Mauseingaben beziehen. Diese Algorithmen sollten anschliessend an verschiedenen Testpersonen, sowie mit verschiedenen Anwendungen getestet und untersucht werden. Dabei stellte ich mir folgende Fragen:

- I. Was ist algorithmische bzw. generative Kunst? Was zeichnet sie aus?
- II. Wie kann durch Algorithmen mit Tastatur - und Mauseingaben ein ansprechendes und interessantes Bild dargestellt werden, welches je nach Programm sowie Testperson unterschiedlich, dabei aber trotzdem persönlich und individuell für die jeweilige Testperson bleibt?
- III. Wie und warum unterscheiden sich die entstehenden Bilder, welche Schlüsse lassen sich daraus ziehen?
- IV. Wie könnte man die vorhandenen Programme im Nachhinein noch überarbeiten?

Ich benutze die Programmiersprache Processing. Diese ist eine objektorientierte, stark typisierte Programmiersprache, welche auf Java basiert. Diese implementierte ich in die Entwicklerumgebung Eclipse und führte die Anwendungen als Applet aus. Dies war von Nöten, um genug Tief in das Betriebssystem eindringen zu können. Diese Methode bot sich an, da ich schon meine Erfahrungen mit diesen beiden Sprachen während meiner Schulzeit gemacht habe und Processing für die Einsatzbereiche Grafik, Simulation und Animation spezialisiert ist.

3. Algorithmische Kunst

In diesem Kapitel werde ich genauer darauf eingehen, was algorithmische Kunst ist, welche Rolle der Algorithmus dabei spielt und wie die algorithmische Kunst in Verbindung mit der algorithmischen Revolution steht. Weiterhin werde ich ein Beispiel der algorithmischen Kunst anschauen und aufzeigen, was es auszeichnet.

Die algorithmische Kunst, auch bekannt unter dem Begriff "computergenerierte Kunst", gehört zu der digitalen Kunst und ist eine Untergruppe der generativen Kunst. Die ersten dokumentierten Beispiele für algorithmische Kunst stammen aus den frühen 1960er Jahren. Damals wurden die Bilder meistens jedoch noch physisch hergestellt und erst mit dem Fortschreiten der algorithmischen Revolution in der Kunst, wanderten die Ergebnisse der Algorithmen auf den Bildschirm.¹

3.1 Beispiel für algorithmische Kunst

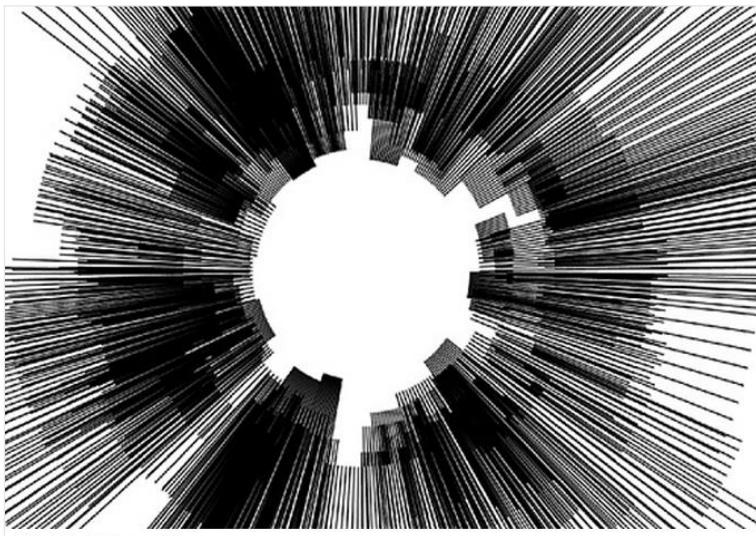


Abb. 1 «Arc Drawings 001», 2011 von Marius Watz.²

Das Bild «Arc Drawings» von Marius Watz ist ein gutes Beispiel für die algorithmische Kunst. Es zeichnet sich dadurch aus, dass es nur aus simplen geometrischen Formen zusammengesetzt ist, hier aus Linien. Diese simplen geometrischen Formen werden dann meistens in Kombination mit weiteren Formen zu etwas Komplexerem zusammengesetzt, hier zu einem Kreis. Weiterhin spielt die Mathematik eine grosse Rolle, denn ein sehr grosser Teil des Programmierens basiert auf ihr. So werden sehr

¹ Vgl. Wikipedia, Digitale Kunst.

² Vgl. mariuswatz.com

oft mathematische Operatoren, Gleichungen, etc. für die Programme verwendet. Hier zum Beispiel sind es wahrscheinlich Vektoren mit unterschiedlichen Längen und Angriffspunkten, welche dann im Kreis immer um eine bestimmte Anzahl Grad gedreht werden. In diesem Bild spielen die Farben keine besondere Rolle, es wurde nur Schwarz benutzt. Meistens jedoch werden bei der algorithmischen Kunst viele unterschiedliche Farben eingesetzt, da diese relativ einfach einzubauen sind und mit der riesigen Farbpalette von 16.8 Millionen Farben auch genügend zur Verfügung stehen. Was auch sehr verbreitet ist, ist der Gebrauch von Transparenzen. Man benutzt also Farben, die nicht ganz deckend sind und sich damit überlagern können. Damit lassen sich nämlich oft spektakuläre Bilder generieren. Das eigentliche Hauptmerkmal der algorithmischen Kunst ist jedoch, dass bei den gleichen Startbedingungen bzw. den gleichen Variablen immer das gleiche Ergebnis entsteht. Dies ermöglicht es viele verschiedene Bilder in relativ kurzer Zeit zu generieren, indem man nur wenige Parameter ändert. Dies ist eines der wichtigsten Merkmale, inwiefern die algorithmische Kunst sich von anderer Kunst unterscheidet.

Algorithmische Kunst wird von den meisten Personen wahrscheinlich nicht immer als schön interpretiert. Dies ist aber auch nicht unbedingt das Ziel der algorithmischen Kunst. Die Kunst ist im Programm oder in der zugrundeliegenden Idee zu suchen.³ Denn meistens sind es nur geometrische Figuren, die auf eine spezielle Art zusammengewürfelt sind. Doch trotzdem wirkt die algorithmische Kunst immer auf eine gleiche, doch aber auch auf eine besondere Art. Auf mich wirkt sie immer geordnet, aufgeräumt, bedacht und ruhig, auch wenn sehr viele Zufallsvariablen verwendet werden. Dies hat wahrscheinlich den Grund, dass meistens sehr viel mehr hinter den Bildern steht, als wir auf den ersten Blick erkennen. Weitere Adjektive, die meiner Meinung nach die algorithmische Kunst sehr gut beschreiben sind: vorbereitet, entspannt, sorglos, gelassen und ausgewogen. Ich habe nie das Gefühl in Stress oder Hektik zu verfallen, wenn ich ein algorithmisches Bild anschau. Besonders ist auch noch, dass algorithmisch generierte Bilder, falls man sie distanziert und von etwas weiter weg betrachtet, anders wirken, als wenn man sie im Detail betrachtet.

³ Vgl. Wikipedia, Generative Kunst.

3.2 Der Algorithmus

Dass eine Arbeit zur algorithmischen Kunst gezählt werden kann, muss in der Entstehung ein Prozess integriert sein, welche auf einem Algorithmus basiert. Ein Algorithmus ist vereinfacht gesagt eine genaue Anweisung, um eine Aufgabe aufzuführen. Der Begriff hat seinen Ursprung in der Mathematik, um Probleme mit einer Schritt-für-Schritt-Prozedur zu lösen. Zum Beispiel sind die gewöhnlichen Verfahren für Multiplikation und Division Algorithmen. Doch es wäre falsch zu sagen, dass nur mathematische Operationen und Verfahren als Algorithmen gezählt werden können. Heute gilt jede genau definierte Methode bzw. Technik als Algorithmus, solange man immer das gleiche Ergebnis bei gleichen Startbedingungen bekommt. Dazu zählen zum Beispiel alle Programme, Anweisungen und Rezepte. So können auch Maschinen programmiert werden, um Anweisungen auszuführen. In der Theorie sollte das programmieren von Maschinen eigentlich immer funktionieren, da es aber mechanische und menschliche Fehler, sowie externe Effekte gibt, ist dem nicht immer so.⁴

3.3 Die algorithmische Revolution

Normalerweise, falls eine Revolution vor uns liegt, kündigt sie sich langsam an. Die algorithmische Revolution dagegen liegt bereits hinter uns und nur wenige haben sie bemerkt. Umso wirkungsvoller ist sie gewesen. Sie begann um 1930 und erreichte 1960 die Kunst.⁵ Inzwischen gibt es fast keinen Bereich unseres Lebens mehr, in welchem nicht irgendein Algorithmus eine Rolle spielt. Der wohl heute bekannteste Gebrauch liegt in der Computerprogrammierung. Ein Programm ist ein Algorithmus, welcher in einer Sprache formuliert ist, die der Computer versteht und umsetzen kann. So kann heute jedes Problem, das programmierbar ist, mit Programmiersprachen algorithmisch gelöst werden. Eine der wahrscheinlich Bekanntesten und Ersten dieser Problemlösung ist das von 1936 von Alan Turing entwickelte Konzept der Turing Maschine. Doch die algorithmische Revolution spielte nicht nur in der Mathematik und Informatik eine grosse Rolle. Sie machte auch keinen Halt vor der Kunst. So wurde mit der Erfindung des Computers eine völlig neue Welt für Künstler eröffnet, welche ihnen erleichterte mit Algorithmen zu arbeiten. Es gab jedoch auch schon früher

⁴ Vgl. Wikipedia, Algorithmus.

⁵ Vgl. ZMK, Die algorithmische Revolution.

algorithmische Kunst, aber diese wurde meistens mit einem Roboter ausgeführt und war dementsprechend umständlich. Durch die Möglichkeit die Kunst auf einem Bildschirm darzustellen, wurde der Durchbruch für die algorithmische Kunst ermöglicht.

4. Generative Kunst

In diesem Kapitel werde ich genauer auf die generative Kunst eingehen und wie im letzten Kapitel durch die Anschauung eines Beispiels die typischen Merkmale aufzeigen.

Wie vorhin schon erwähnt ist die algorithmische Kunst eine Untergruppe der generativen Kunst. In der generativen Kunst steht nicht unbedingt das Kunstwerk oder das Endprodukt im Zentrum, sondern der Entstehungsprozess und die ihm zugrundeliegenden Idee. Das Kunstwerk entsteht durch das Abarbeiten eines vom Künstler geschaffenen Regelsatzes bzw. Programmes. Das Ergebnis bewegt sich zwar in gegebenen Grenzen, ist darin jedoch unvorhersehbar. Generative Kunst wird oft gebraucht, um die algorithmische Kunst zu beschreiben. Doch kann sie auch noch ganz anderen Bereichen zugeordnet werden, wie Chemie, Musik, Biologie und Mathematik.⁶

4.1 Beispiel für generative Kunst



Abb. 2 Das Gemälde «*Untitled*», 1970» von Cy Twombly.

Hier sehen wir ein Bild des Künstlers Cy Twombly. Er wurde unter anderem bekannt für seine Kunst der Kritzeleien, wie dieses Bild eines ist. Dabei malte er auf eine vorbereitete Leinwand irgendwelche Striche, manchmal verband er sich sogar die Augen dabei. Damit können seine Werke der generativen Kunst zugeordnet werden. Sie zeichnen sich dadurch aus, dass alle ähnlich sind, da es eigentlich nur Kritzeleien

⁶ Vgl. Wikipedia, Generative Kunst.

sind. Aber trotzdem sind alle auch unterschiedlich. Weiterhin stand nicht das Kunstwerk im Zentrum, sondern der Weg, wie es entstanden ist. Die Wirkung auf den Menschen ist sehr vergleichbar mit der algorithmischen Kunst, was nicht verwunderlich ist, da die algorithmische Kunst eine Untergruppe der generativen Kunst ist. Cy Twombly ist auch ein gutes Beispiel dafür, dass es auch generative Kunst gibt, welche nicht computergeneriert ist. Man kann ihn auch als Bindeglied zwischen manueller und computergenerierter Kunst sehen. Der Unterschied zwischen der generativen Kunst und den vielen anderen Kunstarten ist, dass das Ergebnis bzw. das Endprodukt nicht im Fokus des Künstlers stehen. Vielmehr das wie und warum, also der Entstehungsprozess und die Idee, die dahintersteckt.

5. Vorgehen

In diesem Kapitel werde ich genau beschreiben, wie ich vorgegangen bin, um meine Bilder zu erzeugen.

Um meine Bilder zu erstellen, benutze ich hauptsächlich die Programmiersprache Processing. Diese Sprache hat ihren Schwerpunkt auf Grafik, Simulation und Animation gelegt und entsprach damit meinen Bedürfnissen. Um jedoch genügend tief ins Betriebssystem eindringen zu können, um Maus und Tastatureingaben nativ aufzunehmen, musste ich Processing in die Entwicklerumgebung Eclipse integrieren. Danach konnte ich mein Programm als Applet ausführen und die nötigen Daten sammeln, um die Bilder zu generieren.

5.1 Grundgerüst

Ich baute mir zuerst ein Grundgerüst zusammen, welches mir dann auch immer als Vorlage für die neuen Programmierungen diente. Ich arbeitete mit verschiedenen Versionen, welche ich dann immer wieder überarbeiten konnte. So entstanden 3 Versionen des Grundgerüstes. Die letzte Version enthält ca. 150 Linien an Code und fasst alle Signale auf, wie der Mensch mit dem Computer kommunizieren kann.

Darunter fällt:

- Tastaturtaste drücken
- Tastaturtaste loslassen
- Maustaste links drücken
- Maustaste links loslassen
- Maustaste rechts drücken
- Maustaste rechts loslassen
- Maustaste mitte drücken
- Maustaste mitte loslassen
- Mausbewegungen
- Mausstillstand
- Mausradaktivität
- Randberührungen

Weiterhin kann es auch die Mausgeschwindigkeit in X und Y Richtung sowie die Schreibgeschwindigkeit bestimmen.

Im Anhang befindet sich der Quellcode der letzten Version des Grundgerüstes.

5.2 Programme

Ausgehend vom Grundgerüst programmierte ich zwölf verschiedene Programme. Jedes Programm untersucht einen anderen Aspekt, wie der Mensch das Geschehen auf dem Bildschirm beeinflussen kann. Wie diese Programme entstanden sind und was sie genau tun, ist im Arbeitsjournal genau dokumentiert. Auch hier arbeitete ich wieder mit verschiedenen Versionen. Je weiter vorgeschritten meine Arbeit war, desto weniger Versionen eines Programmes entstanden, da ich immer genauer wusste, was ich will und wie dies umzusetzen ist. Die Programme unterscheiden sich zum Teil enorm in der Länge und im Zeitaufwand. So ist das Programm 5 eindeutig das Längste und Zeitaufwendigste gewesen, da ich für jede Taste einen eigenen Case machen musste, um die Position eindeutig zu bestimmen. Nur für dieses Programm alleine hatte ich einen Zeitaufwand von mehreren Stunden. Die entstandenen Bilder speicherte ich einmal als PDF, welches vektoriell basiert ist, und einmal als TIF-Datei, um sie später zu analysieren und auszudrucken. Die meisten Programme enthalten drei Farben. Diese konnten von der Testperson im Voraus bestimmt werden. Mit diesem Schritt sind die Bilder persönlicher und individueller geworden, da jeder seinen drei verschiedenen Lieblingsfarben hat. Die Farbpalette, von der die Farben gewählt werden konnten, umfasst den gesamten RGB-Bereich, also 16.8 Millionen Farben.

Im Anhang befindet sich der genaue Quellcode der letzten Version der Programme.

5.3 Testpersonen

Für meine Testreihe wählte ich sechs Personen aus. Ich weiss, dass diese Anzahl nicht ausreicht, um statistische Aussagen tätigen zu können. Doch da ich keine statistischen Aussagen machen wollte, konnte ich mich mit sechs Personen zufriedengeben. Weiterhin ist der Testaufwand mit ungefähr zwei Stunden pro Person enorm gross, was mich leider daran hinderte, weitere Testpersonen aufzunehmen. Ich nahm immer einen Mann und eine Frau, welche ungefähr das gleiche Alter haben. Die Altersabschnitte teilte ich wie folgt ein: 0-20, 20-30, 30+ Jahre. Dies machte ich, um doch noch ein wenig Diversität in meine Testgruppe zu bringen.

5.4 Testparcours

Der Testparcours besteht aus vier Abschnitten. Für jeden Abschnitt hatte die Testperson 15 Minuten Zeit. Und nach jedem Abschnitt wurden die Programme wieder neu gestartet, damit für die verschiedenen Abschnitte auch verschiedene Bilder entstehen. Alle zwölf Programme liefen jeweils gleichzeitig. Vor dem Test musste die Testperson eine Einverständniserklärung unterzeichnen und einen kleinen Steckbrief ausfüllen, welche über Alter und Computerkenntnisse aufklären soll. Die Testreihen wurden bei mir zu Hause durchgeführt an einem 24 Zoll Monitor mit einem Seitenverhältnis von 16:9. Die Bilder haben darum auch alle genau dieses Verhältnis. Als Tastatur stand die Roccat Ryos MK Pro mit dem schweizer Tastaturlayout und den braunen Cherry MX Tasten zur Verfügung. Als Maus diente die Sharkoon Drakonia, eingestellt auf 1500 DPI.

5.4.1 Das Schreiben

Zuerst musste die Testperson einen Text schreiben. Hier wollte ich vor allem die Schreibgeschwindigkeit sowie den unterschiedlichen Tastengebrauch untersuchen. Dazu bekam die Testperson ein von mir vorbereitet Word-Dokument. In diesem Word-Dokument habe ich vier Wörter bzw. Teilsätze vorgegeben, welche im Text vorkommen sollten. Dies machte ich, damit alle die gleiche Ausgangslage haben, aber dennoch nicht völlig frei sind. Es handelt dabei um: «während er ging, ging etwas kaputt», «lächelt», «Der Computer», und die Zahl «55». Die Zahl fügte ich erst später hinzu, da ich noch untersuchen wollte, ob die Testpersonen eher den Nummernblock rechts oder die Zahlenreihen oben auf der Tastatur benutzen. Die entstandenen Texte waren meist recht amüsant.

5.4.2 Das Zeichnen

Danach musste die Testperson ein Bild am Computer mit Microsoft Paint zeichnen. Ich gab ihr auch wieder Wörter vor, an denen Sie sich orientieren soll. Es handelt sich um folgende Wörter: «Rechnen», «Emotion» und «gesellschaftlich». Hier sollte untersucht werden, welche Bilder die Programme erstellen mit den Informationen, die sie von der Testperson bekommen, welche gerade selbst ein Bild malt. Die Grösse des Bildes, welches die Testperson selbst macht, konnte Sie hier selber bestimmen. Doch alle haben die vorgegebene Grösse von Paint benutzt.

5.4.3 Das Surfen

Im dritten Teil der Testreihe hatte die Testperson 15 min Zeit im Internet zu surfen. Sie war hier völlig frei und konnte tun, was Sie wollte. Darum unterscheiden sich die entstandenen Bilder hier auch am stärksten.

5.4.4 Das Spielen

Im letzten Abschnitt spielte die Testperson das Spiel «Age of Empires III». Dies ist ein Strategiespiel, in welchem es darum geht eine Siedlung von Grund auf aufzubauen und durch verschiedene Zeitalter zu führen, um am Schluss den Feind in einer Schlacht zu besiegen. Ich wählte dieses Spiel, weil die Grundprinzipien des Spiels einfach zu erlernen sind, es aber trotzdem auch geübte Spieler fordern kann. Weiterhin fand ich die Spielmechanik spannend, also wie der Spieler das Geschehen auf dem Bildschirm steuern kann. Dies kann man in diesem Spiel nämlich auf viele verschiedene Arten, wenn man es mit anderen Spielen vergleicht. Dies ist wiederum ein Vorteil für meine Arbeit, da viele verschiedenen Bilder erzeugt werden können.

5.5 Ergebnisse

Die von den Programmen erzeugten Bilder befinden sich im Anhang in 9.3 auf Seite 95.

Die von den Testpersonen gemalten Bilder und geschriebenen Texte befinden sich im Anhang ab 9.4 auf Seite 143.

6. Analysen

In diesem Kapitel werde ich auf einzelne Bilder eingehen und diese analysieren. Weiterhin werde ich meine Programme reflektieren und festhalten wie diese noch verbessert werden können.

6.1 Bildanalysen

Da insgesamt 288 Bilder erstellt wurden, werde ich nicht auf jedes einzelne eingehen können. Ich werde mich auf einige wenige Bilder beschränken, welche meiner Meinung nach erwähnenswerte Besonderheiten aufweisen. Im Anhang sind alle restlichen Werke zu finden.

6.1.1 Programm 1, Spiel, Simon



Abb. 3

Dieses Bild habe ich nicht gewählt, weil ich es besonders schön oder ansprechend finde. Der Grund ist hier ein ganz anderer. Wenn man das Bild genau anschaut, bemerkt man, dass sich ein X über das Bild zieht. Nur das rechte obere Eck ist ein wenig schwach ausgeprägt, aber dennoch ist es zu erkennen. Nun, wie ist es zu dieser besonderen Form gekommen? Das Bild zeichnete die Mausbewegungen während des Spiels Age of Empires 3 auf. In diesem Spiel wird die Maus unter anderem dafür verwendet, um den sichtbaren Bereich des Spielfelds anzupassen. Das heisst, man

bewegt die Maus an den Rand des Bildschirmes und im Spiel bewegt sich das Sichtfeld dann in diese Richtung. Dieses X bildet sich dann aufgrund von diesen Bewegungen. Man kann dieses Mausverhalten auch bei anderen Spielen beobachten, bei welchen man von oben auf das Spielgeschehen blickt, wie Dota 2 oder League of Legends. Doch das aussergewöhnliche daran ist, dass das X stärker ausgeprägt ist, je erfahrener der Spieler ist. Denn durch die schräge Bewegung spart man Weg und Zeit, anstatt sich nur in der Horizontale und Vertikale zu bewegen. Das Prinzip ist vergleichbar mit einem rechtwinkligen Dreieck. Die Hypotenuse ist kürzer als die Ankathete und die Gegenkathete zusammen. Und auf genau diese Einsparung sind erfahrene Spieler aus. Denn in hitzigen Onlinegefechten oder auch bei höheren Schwierigkeitsstufen zählt jede Sekunde und kann am Schluss über Sieg oder Niederlage entscheiden. Wie man im Anhang nachschauen kann, ist dieses das einzige Bild, welches diese Besonderheit aufweist. Denn diese Testperson ist auch die Einzige, welche vor dem Test schon viele Spielstunden Erfahrungen gesammelt hat.

6.1.2 Programm 5, Kyo, Aufsatz

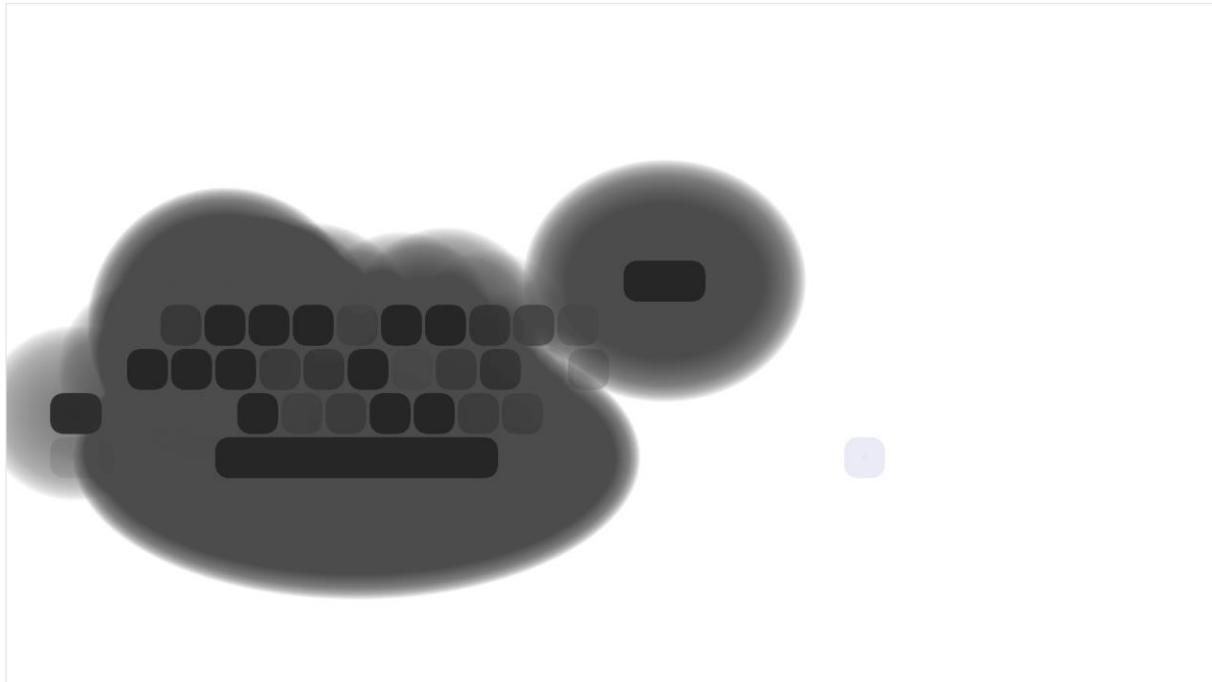


Abb. 4

Auch dieses Bild habe ich nicht gewählt, weil ich es optisch besonders ansprechend finde. Es sind ja auch eigentlich nur ein paar schwarze Flächen. Wie auch schon vorhin liegt der Grund etwas tiefer. Dieses Bild ist entstanden, während die Testperson einen Aufsatz geschrieben hat. Dabei zählte das Programm, wie oft jede einzelne Taste gebraucht worden war, und zeichnete die Tasten im gewohnten Tastaturmuster auf

das Bild. Je öfter die Taste dabei losgelassen wurde, desto kräftiger bzw. dunkler wird ihr Abdruck dabei. Zudem zeichnete das Programm noch eine Art Schatten um die jeweiligen Tasten, dieser wird bei mehrfacher Benutzung grösser. So kann man jetzt sehr gut ablesen, welche Tasten diese Testperson oder auch allgemein am meisten in der deutschen Sprache benutzt werden. Der Grund, warum ich genau dieses Bild genommen habe und nicht ein anderes, ist, weil dadurch, dass die Testperson vergleichsweise wenig geschrieben hat und die Farbe Schwarz gewählt hat, die einzelnen Tasten noch zu sehen sind und nicht im Schatten verschwinden, wie es bei anderen der Fall ist. Im Bild stechen als Erstes die Leertaste, die Rücktaste (Backspace) und die E-Taste heraus. Dies sind die mit Abstand am meistbenutzten Tasten in diesem Bild. Danach folgt noch die Shift-taste, doch diese auch schon mit grösserem Abstand. Danach wird die Erkennung zunehmend schwieriger, da die Tasten nahe beieinanderliegen und nicht mehr gross herausstechen. Trotzdem würde ich sagen, dass nach der Shift-taste die R, T, U, I, A, S, D, H, C, N und M Tasten folgen, alle mit ungefähr der gleichen Häufigkeit. Danach mit noch weniger Benutzung die W, Z, O, P, F, G, J, K, L, Ü, Ä, V, B, Komma, Punkt und linke CTRL Taste, sowie eine Pfeiltaste, doch ich kann nicht genau sagen welche. Wahrscheinlich hat die Testperson die Tastenbefehle CTRL+C und CTRL+V oft gebraucht und weicht darum etwas ab. Eigentlich nicht gebraucht wurden hingegen die Q, Ö, Y, X Tasten, sofern wir nur das erweiterte Alphabet anschauen. Spannend wäre jetzt hier auch noch anzuschauen, ob und wie sich die Bilder bei verschiedenen Sprachen unterscheiden, oder eben nicht.

6.1.3 Programm 7, Julia, Internet

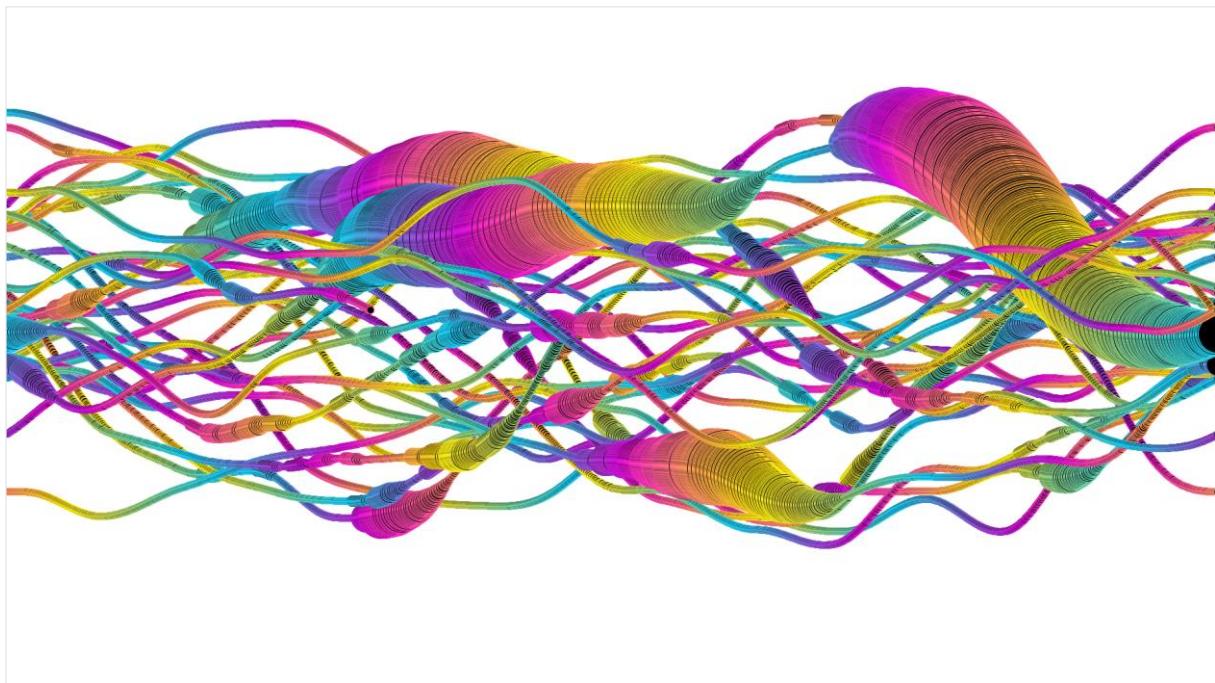


Abb. 5

Bei diesem Bild finde ich die Farbkombination sehr gelungen. Wahrscheinlich hat die Testperson sehr darauf geachtet, dass die Farben auch in Kombination gut zusammenpassen und nicht nur einzeln schön aussehen. Im Gegensatz zu den vorherigen Bildern finde ich dieses dafür in der Entstehung etwas langweiliger. Bei diesem Programm wandert ein Punkt von links nach rechts und hinterlässt eine Linie, in der Höhe kann der Punkt variieren. Falls man das Mausrad benutzt wird die Linie dicker. Sobald das Mausrad nicht mehr benutzt wird, wird die Linie wieder dünner. So zeigt die Serie von diesem Programm vor allem im Internet schöne Besonderheiten, da dort das Mausrad am meisten benutzt wird. Bei anderen Anwendungen dafür aber eher nicht. Ich stelle mir bei diesem Bild immer ein Wurm vor, welcher an einigen Stellen dicker, an einigen dünner ist und in vielen schönen Farben strahlt.

6.1.4 Programm 8, Spiel, Julia



Abb. 6

Bei diesem Bild liegt die Besonderheit in der Persönlichkeit, welche es erstellt hat. Dieses Programm zeichnet gerade Linien, falls die Maus gedrückt und bewegt wird, vom Anfangspunkt, wo die Maustaste gedrückt wird, bis zum Schlusspunkt, wo die Maustaste wieder losgelassen wird. Das Bild ist während des Spielens entstanden. Dort wird die vorher beschriebene Geste benutzt, um mehrere Einheiten auf einmal zu markieren. Man muss nämlich ein Viereck um sie ziehen. Doch das nun wirklich Besondere an diesem Bild ist, dass alle Linien von oben links nach unten rechts verlaufen, also alle in die gleiche Richtung. Das heisst, die Person hat immer genau die gleiche Geste benutzt, um ein Viereck zu ziehen, oft sogar noch von der ungefähr selben Grösse. Fast nie hat sie von rechts nach links gezogen. Bei den anderen Testpersonen ist das Bild viel unstrukturierter und es gibt auf keinen anderen so vielen Linien in dieselbe Richtung. Ich finde, dass diese Angewohnheit der Testperson etwas sehr Spezielles ist und es damit verdient hat, erwähnt zu werde.

6.2 Reflexion

In diesem Unterthema werde ich zuerst beschreiben, wie die von mir geschriebenen Programme in Zukunft noch verbessert werden können. Ich werde allgemeine Verbesserungen erwähnen, welche alle Programm betreffen und danach werde ich

auf die einzelnen Programme genauer eingehen. Zum Schluss wurde ich noch eine persönliche Reflexion abgeben.

6.2.1 Allgemein

- **Farben**

Ich würde nicht mehr die ganze RGB Farbpalette zur freien Verfügung stellen, sondern nur noch Farben zur Auswahl geben, welche gut zueinanderpassen und sich gut für die Programme eignen.

- **Mehrere Programme gleichzeitig starten**

Man könnte die einzelnen Programme umschreiben und zusammenfügen, damit alle miteinander gestartet werden können.

- **Programmcode**

Der Programmcode an sich kann noch optimiert werden, da Personen, die den Code nicht geschrieben haben, wahrscheinlich sehr viel Mühe haben werden, ihn zu verstehen

- **Speichern**

Ich würde das Speichern automatisieren, damit die Programme sich nach einer gewissen Zeit selbst schliessen und die Ergebnisse speichern.

- **Mehrere Testpersonen**

Man könnte mehrere Testpersonen verwenden, um bessere Vergleiche mit Geschlechter und Alter zu machen.

6.2.2 Einzelne Programme

- **Programm 1**

Die Übergänge zwischen den schnellen und den langsamen Bewegungen könnte noch schöner und flüssiger gemacht werden.

- **Programm 2**

Die Übergänge zwischen den stärkeren und den schwächeren Farben eines Punktes kann man noch schöner gestalten.

- **Programm 3**

Man müsste schauen, dass alle verwendeten Farben gleich hell sind, damit man besser Schlüsse daraus ziehen kann.

- **Programm 4**

Die Übergänge zwischen den stärkeren und den schwächeren Farben eines

Punktes kann man noch schöner gestalten. Kann mehr Farben einsetzen, um das Bild spannender zu gestalten.

- **Programm 5**

Die Schatten der Tasten dürfen die eigentlichen Tasten nicht überlagern. Man sollte ein besseres Verhältnis zum dunkler werden finden, oder bessere Farben wählen, bei denen es nicht so stark auftritt.

- **Programm 6**

Man könnte das Programm umschreiben, dass die Striche nicht zur Mitte hinführen, sondern dorthin, von wo die Maus kommt.

- **Programm 7**

Sollte schauen, dass es auch nach oben und unten über den Bildschirmrand hinausgeht und am Bildschirmrand nicht diese schwarzen Kreise gibt.

- **Programm 8**

Eigentlich gut, könnte noch schauen, dass die Striche erst ab einer bestimmten Länge gezeichnet werden.

- **Programm 9**

Eigentlich gut, braucht aber noch eine Lösung, falls Punkte über den Bildschirm hinauswachsen würden.

- **Programm 10**

Braucht eine bessere Spirale, damit es bei vielen Eingaben nicht unübersichtlich wird und nicht so schnell über den Bildschirmrand hinauswächst.

- **Programm 11**

Eigentlich gut, könnte noch den Abstand, der für die Verbindung nötig ist, erhöhen, damit es beim Bilderzeichnen nicht unübersichtlich wird.

- **Programm 12**

Man sollte mehr Farbe einsetzen, damit das Bild spannender wird.

6.2.3 Persönliche Reflexion

Natürlich gibt es Sachen, die mir gut gelungen sind und auch welche, die ich das nächste Mal besser machen könnte. In diesem Unterkapitel behandle ich diese.

Gut gelungen ist mir sicher das Zeitmanagement. Ich kam nie in den Stress und war schon sehr früh mit dem Grossteil der Arbeit fertig und hatte damit noch genug Zeit, um den schriftlichen Teil fertigzustellen. Etwas, was sehr direkt mit dem vorhin erwähnten zusammenhängt, ist meine Themenwahl. Da das Programmieren mir sehr

liegt und auch Spass bereit, hatte ich keine Probleme zahllose Stunden und freie Nachmittage in meine Maturarbeit zu investieren. Weiterhin sind auch alle Testläufe mit den Testpersonen gut gelaufen und es gab keine Probleme mit den Programmen. Auch bin ich stolz auf meine selbst geschriebenen Programme und die Bilder, welche sie erzeugt haben. Auch wenn, falls eine fremde Person die Programme durchlesen sollte, sie wahrscheinlich nicht verstehen wird, da sie nicht sehr *schön* geschrieben sind.

Würde ich noch einmal von ganz vorne beginnen, würde ich mehr auf Qualität und weniger auf Quantität setzen. Jetzt entstanden über 200 Bilder und ich schrieb zwölf Programme. Ich kann mir gut vorstellen, dass ich vielleicht nur noch drei Programme schreiben würde, diese aber jedoch einiges komplexer sein würde. Weiterhin würde ich wahrscheinlich auch die Testpersonen weglassen und nur noch mich als Tester brauchen, da ich so viel Zeit sparen kann. Denn alle Testläufe mussten bei mir zu Hause stattfinden, da die Tests ziemlich viel Rechenleistung brauchen. Ich versuchte es mit einer Testperson auf einem Notebook und die Ergebnisse waren zum Teil nicht mehr wirklich vergleichbar. Auch habe ich vor mich auf eine bestimmte Programmart zum Testen festzulegen und nicht mehr so wie jetzt vier verschiedene Anwendungen. Hier würde ich wahrscheinlich auf Computerspiele zurückgreifen, da diese meiner Meinung nach die spannendsten Ergebnisse abliefern und ich ein leidenschaftlicher Spieler bin.

7. Fazit

Das Thema meiner Arbeit war es, ausgehend von Maus und Tastaturbewegungen algorithmische Bilder entstehen zu lassen, welche sich nach Anwendung und Person unterscheiden. Der eigentliche Sinn und warum diese Arbeit in den Bereich des bildnerischen Gestaltens gehört, war zu Beginn noch ein wenig unklar. Doch dies kristallisierte sich mit dem Voranschreiten der Arbeit heraus.

Um die Bilder zu erzeugen, programmierte ich zwölf Programme in der Entwicklerumgebung Eclipse und testete diese an sechs Testpersonen. Ich benutze die Programmiersprache von Processing in Verbindung mit Java. Jedes Programm konzentrierte sich auf einen anderen Aspekt, wie der Mensch das Geschehen auf dem Bildschirm beeinflussen kann. Zum Schluss kamen über 3500 Zeilen an Code zusammen. Aus den Testläufen entstanden gesamt 288 Bilder. Sie gehen zum Teil weit auseinander und unterscheiden sich völlig. Andere sind zum Teil recht ähnlich und vergleichbar und wenige sind sogar leider weiss geblieben.

Falls meine Arbeit noch weitergehen würde, hätte ich vor, ausser die selbst geschriebenen Programme noch zu verbessern, andere Betriebssysteme und Monitorgrößen zu vergleichen. Ich könnte mir sogar vorstellen den Desktop zu verlassen, und in den mobilen Bereich zu wechseln. Denn dort wartet mit dem Touchscreen und noch unterschiedlicheren Betriebssystemen sicher Spannendes auf mich. Ansonsten könnte ich noch weitere Spiele untersuchen, da diese meiner Meinung nach die spannendsten Ergebnisse liefern. Weiterhin könnten mehrere Eingaben in einem Programm kombiniert werden, um noch faszinierende Bilder zu erschaffen.

Meine Maturarbeit war ein anstrengendes und zeitraubendes Projekt. Doch sie hat mir auch viel Spass bereitet. Vor allem die Testläufe waren meistens sehr amüsant. Und da mir das Programmieren auch liegt, kann ich positiv auf diese Arbeit zurückblicken. Nur den Aufwand die vielen Bilder zum Schluss gescheit zusammenzustellen, habe ich ziemlich unterschätzt. Auch falls ein Programm gerade nicht machte, was es sollte, vergingen einige Momente der Frustration. Doch schlussendlich habe ich es geschafft und einiges dabei gelernt. Das kreative Arbeiten, das Programmieren, die Testläufe, alles hat mir enorm Freude bereitet und ich bin stolz auf das, was ich erreicht habe.

8. Quellenverzeichnis

(10. Juni 2016). Von <http://kra.lc/blog/2016/02/java-global-system-hook/> abgerufen

(23. 07 2016). Von Wikipedoa: <https://de.wikipedia.org/wiki/Perlin-Noise> abgerufen

America Pink. (29. Mai 2016). Von http://america.pink/algorithmic-art_330275.html abgerufen

Casey Reas, B. F. (2001). *Processing: a programming handbook for visual designers and artists*. Massachusetts Institute of Technology.

kristian. (24. 07 2016). *Github*. Von <https://github.com/kristian/system-hook> abgerufen

Verostko. (29. Mai 2016). Von <http://www.verostko.com/algorithm.html> abgerufen

Wikipedia. (29. Mai 2016). Von https://de.wikipedia.org/wiki/Digitale_Kunst abgerufen

Wikipedia. (29. Mai 2016). Von https://de.wikipedia.org/wiki/Generative_Kunst abgerufen

Wikipedia. (29. Juli 2016). Von Processing: <https://de.wikipedia.org/wiki/Processing> abgerufen

ZMK. (29. Mai 2016). Von <http://www01.zkm.de/algorithmische-revolution/> abgerufen

9. Anhang

9.1 Grundgerüst

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Stable3 extends PApplet

{
    public static void main(String[] args) {
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Stable3());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
    
```

```

    public void keyPressed(GlobalKeyEvent event) {
        switch (event.getVirtualKeyCode()) {
            case 191:
                exit = true;
                break;
        }
        // System.out.println(event);
    }

    @Override
    public void keyReleased(GlobalKeyEvent event) {
        // System.out.println(event);
    }
);

GlobalMouseHook mouseHook = new GlobalMouseHook();
mouseHook.addMouseListener(new GlobalMouseAdapter() {
    @Override
    public void mousePressed(GlobalMouseEvent event) {
        // System.out.println(event);
        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_LEFT) != GlobalMouseEvent.BUTTON_NO) {
            // println("links");
        }
        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_RIGHT) != GlobalMouseEvent.BUTTON_NO) {
            // println("rechts");
        }
        if (event.getButton() ==
GlobalMouseEvent.BUTTON_MIDDLE) {
            // println("middle");
        }
    }
    @Override
    public void mouseReleased(GlobalMouseEvent event) {
        // System.out.println(event);
    }
    @Override
    public void mouseMoved(GlobalMouseEvent event) {
        // System.out.println(event);
    }
    @Override
    public void mouseWheel(GlobalMouseEvent event) {
        // System.out.println(event);
    }
});

```

```

}

static int x, y;
static float a, b;

Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss ");

static boolean exit = false;

public void settings() {

    fullScreen();

}

public void setup() {

    beginRecord(PDF, "test " + ft.format(date) + ".pdf");
    background(255);
    smooth();

}

public void draw() {

    float x = MouseInfo.getPointerInfo().getLocation().x; /* always
1 */
    float y = MouseInfo.getPointerInfo().getLocation().y; /* always
1 */
    if (x == 1919 || x == 0 || y == 0 || y == 1079) { /* Ränder */

    }
    float speed = abs(x - a) + abs(y - b);
    if (exit == true) {
        endRecord();
        saveFrame("test " + ft.format(date) + ".tif");
        exit();
    }

    a = x; /* always last */
    b = y; /* always last */
}
}

```

9.2 Programme

9.2.1 Programm 1

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;

```

```

import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Programmable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(52000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programmable());

        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;
                }
            }
        });
    }
}

```

```

        });

    }

    int mode = 2;

    int ab = 0;
    int ac = 0;
    int ad = 128;

    int ae = 0;
    int af = 191;
    int ag = 255;

    int ah = 33;
    int ai = 33;
    int aj = 33;

    int ak = ae;
    int al = af;
    int am = ag;

    /*
     * int color1 = color(ab,ac,ad); int color2 = color(ae,af,ag); int
color3 =
     * color(ah,ai,aj); int colorfinal = color(ak,al,am);
    */

    static int x, y;
    static float a, b;
    Date date = new Date();
    SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss ");
    static boolean exit = false;

    public void settings() {

        fullScreen();

    }

    public void setup() {
        beginRecord(PDF, "Programml Kyo Spiel " + ft.format(date) + ".pdf");
        background(255);
        smooth();
        stroke(255);
        strokeWeight(10);

    }

    public void draw() {
        float x = MouseInfo.getPointerInfo().getLocation().x; /* always
1 */
        float y = MouseInfo.getPointerInfo().getLocation().y; /* always
1 */
        float speed = abs(x - a) + abs(y - b);
        float strokeweight = 10 - (speed / 10);/* dist(x,y,a,b) */
        if (strokeweight < 1) {
            strokeWeight(1);
        } else {
            strokeWeight((strokeweight));
        }
    }
}

```

```

    if (exit == true) {
        endRecord();
        saveFrame("Programml Kyo Spiel " + ft.format(date) + "
.tif");
        exit();
    }

    line(x, y, a, b);

    if (mode == 1) {
        if (ak < ab && ak != ab) {
            ak++;
        } else if (ak != 0) {
            ak--;
        }

        if (al < ac && al != ac) {
            al++;
        } else if (al != 0) {
            al--;
        }
        if (am < ad && am != ad) {
            am++;
        } else if (am != 0) {
            am--;
        }
        if ((ak == ab || ak == ab + 1 || ak == ab - 1) && (al ==
ac || al == ac - 1 || al == ac + 1)
            && (am == ad || am == ad + 1 || am == ad - 1))
    {
        mode = 2;
    }
}
if (mode == 2) {
    if (ak < ae) {
        ak++;
    } else if (ak != 0) {
        ak--;
    }

    if (al < af) {
        al++;
    } else if (al != 0) {
        al--;
    }
    if (am < ag) {
        am++;
    } else if (am != 0) {
        am--;
    }
    if ((ak == ae || ak == ae + 1 || ak == ae - 1) && (al ==
af || al == af - 1 || al == af + 1)
        && (am == ag || am == ag + 1 || am == ag - 1))
{
    mode = 3;
}

if (mode == 3) {
    if (ak < ah) {
        ak++;
    } else if (ak != 0) {

```

```

        ak--;
    }

    if (al < ai) {
        al++;
    } else if (al != 0) {
        al--;
    }
    if (am < aj) {
        am++;
    } else if (am != 0) {
        am--;
    }
    if ((ak == ah || ak == ah + 1 || ak == ah - 1) && (al ==
ai || al == ai - 1 || al == ai + 1)
        && (am == aj || am == aj + 1 || am == aj - 1))
{
    mode = 1;
}
stroke(ak, al, am);

a = x; /* always last */
b = y; /* always last */

}
}

```

9.2.2 Programm 2

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;

```

```

import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Programm2stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(50000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm2stable());

        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;

                }
                // System.out.println(event);
            }
        });

        GlobalMouseHook mouseHook = new GlobalMouseHook();
        mouseHook.addMouseListener(new GlobalMouseAdapter() {
            @Override
            public void mousePressed(GlobalMouseEvent event) {
                // System.out.println(event);
                if ((event.getButtons() &
GlobalMouseEvent.BUTTON_LEFT) != GlobalMouseEvent.BUTTON_NO) {
                    click = 1;
                    // println("links");
                }

                if ((event.getButtons() &
GlobalMouseEvent.BUTTON_RIGHT) != GlobalMouseEvent.BUTTON_NO) {
                    click = 2;
                    // println("rechts");
                }
            }
        });
    }
}

```

```

        if (event.getButton() ==
GlobalMouseEvent.BUTTON_MIDDLE) {
            click = 3;
            // println("middle");
        }
    }

}) ;

}

static int x, y;
static float a, b;
int i;
Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss");
int color1 = color(0, 0, 128, 1);
int color11 = color(0, 0, 128, 2);
int color111 = color(0, 0, 128);

int color2 = color(0, 191, 255, 1);
int color22 = color(0, 191, 255, 2);
int color222 = color(0, 191, 255);

int color3 = color(33, 33, 33, 1);
int color33 = color(33, 33, 33, 2);
int color333 = color(33, 33, 33);

static int click = 0;
static boolean exit = false;

public void settings() {

    fullScreen();
}

public void setup() {

    beginRecord(PDF, "Programm2 Kyo Spiel " + ft.format(date) + "
.pdf");
    background(255);
    smooth();
    noStroke();
}

public void draw() {

    float x = MouseInfo.getPointerInfo().getLocation().x; /* always
1 */
    float y = MouseInfo.getPointerInfo().getLocation().y; /* always
1 */

    if (exit == true) {
        endRecord();
        saveFrame("Programm2 Kyo Spiel " + ft.format(date) + "
.tif");
        exit();
    }
    switch (click) {
}
}

```

```

    case 1:
        fill(color1);
        for (int i = 0; i < 150; i++) {

            ellipse(x, y, i, i);
            if (i > 10 && i < 50) {
                fill(color11);
                ellipse(x, y, i, i);
            }
        }
        fill(color111);
        ellipse(x, y, 10, 10);
        click = 0;
        break;
    case 2:
        fill(color2);
        for (int i = 0; i < 150; i++) {

            ellipse(x, y, i, i);
            if (i > 10 && i < 50) {
                fill(color22);
                ellipse(x, y, i, i);
            }
        }
        fill(color222);
        ellipse(x, y, 10, 10);
        click = 0;
        break;
    case 3:
        fill(color3);
        for (int i = 0; i < 150; i++) {

            ellipse(x, y, i, i);
            if (i > 10 && i < 50) {
                fill(color33);
                ellipse(x, y, i, i);
            }
        }
        fill(color333);
        ellipse(x, y, 10, 10);
        click = 0;
        break;
    }
}

```

9.2.3 Programm 3

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;

```

```

import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import processing.opengl.*;
import javax.swing.*;
import processing.opengl.*;
import processing.dxf.*;
import java.util.*;
}

public class Programm3stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(48000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }

        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm3stable());

        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;
                        break;

                }
            }
        });
    }
}

```

```

        key = true;
    }

    @Override
    public void keyReleased(GlobalKeyEvent event) {
        // System.out.println(event);
        long endTime = System.currentTimeMillis();

        float estimatedTime = abs(startTime - endTime);
        startTime = System.currentTimeMillis();
        zeit = estimatedTime;
    }
});

int mode = 1;
static long startTime = 0;

float a = 0;
float b = 0;
float c = 128;

float d = 0;
float e = 191;
float f = 255;

float g = 33;
float h = 33;
float i = 33;

float ae = 100;
static float zeit;

static boolean key = false;
Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss");
static boolean exit = false;

public void settings() {

    fullScreen(P3D);
}

public void setup() {

    beginRaw(DXF, "Programm3 Kyo Spiel " + ft.format(date) + "
.dxf");
    // beginRaw(PDF, "test " + ft.format(date) + ".pdf");

    background(255);
    smooth();

    // strokeWeight(5);
    noStroke();
}

public void draw() {
}

```

```

if (exit == true) {
    endRaw();

    saveFrame("Programm3 Kyo Spiel " + ft.format(date) + "
.tif");
    exit();
}

if (key == true) {
    switch (mode) {
        case 1:
            float ab = 0;
            float ac = 0;
            float ad = 0;

            ab = (ab + zeit) / (a / 255);
            ac = (ac + zeit) / (b / 255);
            ad = (ad + zeit) / (c / 255);
            if (ab > a) {
                ab = a;
            }
            if (ac > b) {
                ac = b;
            }
            if (ad > c) {
                ad = c;
            }
            int color1 = color(ab, ac, ad, ae);
            fill(color1);
            mode = 2;
            break;
        case 2:
            ab = 0;
            ac = 0;
            ad = 0;

            ab = (ab + zeit) / (d / 255);
            ac = (ac + zeit) / (e / 255);
            ad = (ad + zeit) / (f / 255);
            if (ab > d) {
                ab = d;
            }
            if (ac > e) {
                ac = e;
            }
            if (ad > f) {
                ad = f;
            }
            int color2 = color(ab, ac, ad, ae);
            fill(color2);
            mode = 3;
            break;
        case 3:
            ab = 0;
            ac = 0;
            ad = 0;

            ab = (ab + zeit) / (g / 255);
            ac = (ac + zeit) / (h / 255);
            ad = (ad + zeit) / (i / 255);
            if (ab > g) {
                ab = g;
            }
    }
}

```

```

        if (ac > h) {
            ac = h;
        }
        if (ad > i) {
            ad = i;
        }
        int color3 = color(ab, ac, ad, ae);
        fill(color3);
        mode = 1;
        break;
    }

    if (zeit == 0) {
        switch (mode) {
            case 1:
                fill(a, b, c, ae);
                break;
            case 2:
                fill(d, e, f, ae);
                break;
            case 3:
                fill(g, h, i, ae);
                break;
        }
        translate(random(-50, 1919 + 50), random(-50, 1079 + 50),
random(-1000, 500));
        box(100);

        key = false;
    }
}

}

```

9.2.4 Programm 4

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;

```

```

import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Programm4stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(46000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm4stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;
                }
                // System.out.println(event);
            }
        });

        GlobalMouseHook mouseHook = new GlobalMouseHook();
        mouseHook.addMouseListener(new GlobalMouseAdapter() {

            @Override
            public void mouseMoved(GlobalMouseEvent event) {

                if (mousemove < 0) {
                    circle = true;
                }
            }
        });
    }
}

```

```

        mousemove = 20;

        // System.out.println(event);
    }

});

}

int color1 = color(0, 0, 0, 1);
int color11 = color(0, 0, 0, 2);
int color111 = color(0, 0, 0);

double c;
static float d = 0;
static int mousemove = 50;
Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss");

static boolean exit = false;
static boolean circle = false;

public void settings() {

    fullScreen();

}

public void setup() {

    beginRecord(PDF, "Programm4 Kyo Spiel " + ft.format(date) + "
.pdf");
    background(255);
    smooth();
    noStroke();
    // fill(0, 0, 0, 100);

}

public void draw() {

    /*
    float x = MouseInfo.getPointerInfo().getLocation().x; /* always
    float y = MouseInfo.getPointerInfo().getLocation().y; /* always
    */
    if (mousemove < 0) {

        d += 0.1;

    }
    if (circle == true) {

        c = (30 * -Math.exp(-0.1 * d) + 30) * Math.pow(d, 0.333);

        float g = (float) c;
        fill(color1);
        for (int i = 0; i < g; i++) {

            ellipse(x, y, i, i);
            if (i > 10 && i < 50) {
                fill(color11);
                ellipse(x, y, i, i);

    
```

```

        }
    }
    if(g>50) {
        fill(color111);
        ellipse(x, y, g/25, g/25);
    }

    ellipse(x, y, g, g);

    d = 0;
    circle = false;
}
mousemove--;

if (exit == true) {
    saveFrame("Programm4 Kyo Spiel " + ft.format(date) + ".tif");
    endRecord();
    exit();
}

}

```

9.2.5 Programm 5

```
import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.TS_DOWN;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_CONTROL;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_LCONTROL;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_LMENU;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_LSHIFT;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_MENU;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_RCONTROL;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_RMENU;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_RSHIFT;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_RWIN;
import static lc.kra.system.keyboard.event.GlobalKeyEvent.VK_SHIFT;
import java.util.List;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.CopyOnWriteArrayList;
import java.util.concurrent.LinkedBlockingQueue;
import lc.kra.system.LibraryLoader;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
```

```

import lc.kra.system.keyboard.event.GlobalKeyListener;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Programm5stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(44000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm5stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;
                        break;

                }
            }

            @Override
            public void keyReleased(GlobalKeyEvent event) {
                // System.out.println(event);
                switch (event.getVirtualKeyCode()) {
                    case 27:
                        taste = 27;
                        break;
                    case 112:
                }
            }
        });
    }
}

```

```
        taste = 112;
        break;
case 113:
        taste = 113;
        break;
case 114:
        taste = 114;
        break;
case 115:
        taste = 115;
        break;
case 116:
        taste = 116;
        break;
case 117:
        taste = 117;
        break;
case 118:
        taste = 118;
        break;
case 119:
        taste = 119;
        break;
case 120:
        taste = 120;
        break;
case 121:
        taste = 121;
        break;
case 122:
        taste = 122;
        break;
case 123:
        taste = 123;
        break;
case 44:
        taste = 44;
        break;
case 111:
        taste = 111;
        break;
case 106:
        taste = 106;
        break;
case 109:
        taste = 109;
        break;
case 19:
        taste = 19;
        break;
case 144:
        taste = 144;
        break;
case 145:
        taste = 145;
        break;
case 49:
        taste = 49;
        break;
case 50:
        taste = 50;
        break;
case 51:
```

```
        taste = 51;
        break;
case 52:
        taste = 52;
        break;
case 53:
        taste = 53;
        break;
case 54:
        taste = 54;
        break;
case 55:
        taste = 55;
        break;
case 56:
        taste = 56;
        break;
case 57:
        taste = 57;
        break;
case 48:
        taste = 48;
        break;
case 219:
        taste = 219;
        break;
case 221:
        taste = 221;
        break;
case 8:
        taste = 8;
        break;
case 45:
        taste = 45;
        break;
case 36:
        taste = 36;
        break;
case 33:
        taste = 33;
        break;
case 9:
        taste = 9;
        break;
case 81:
        taste = 81;
        break;
case 87:
        taste = 87;
        break;
case 69:
        taste = 69;
        break;
case 82:
        taste = 82;
        break;
case 84:
        taste = 84;
        break;
case 90:
        taste = 90;
        break;
case 85:
```

```
        taste = 85;
        break;
case 73:
        taste = 73;
        break;
case 79:
        taste = 79;
        break;
case 80:
        taste = 80;
        break;
case 186:
        taste = 186;
        break;
case 192:
        taste = 192;
        break;
case 13:
        taste = 13;
        break;
case 46:
        taste = 46;
        break;
case 35:
        taste = 35;
        break;
case 34:
        taste = 34;
        break;
case 103:
        taste = 103;
        break;
case 104:
        taste = 104;
        break;
case 105:
        taste = 105;
        break;
case 107:
        taste = 107;
        break;
case 20:
        taste = 20;
        break;
case 65:
        taste = 65;
        break;
case 83:
        taste = 83;
        break;
case 68:
        taste = 68;
        break;
case 70:
        taste = 70;
        break;
case 71:
        taste = 71;
        break;
case 72:
        taste = 72;
        break;
case 74:
```

```
        taste = 74;
        break;
case 75:
        taste = 75;
        break;
case 76:
        taste = 76;
        break;
case 222:
        taste = 222;
        break;
case 220:
        taste = 220;
        break;
case 223:
        taste = 223;
        break;
case 100:
        taste = 100;
        break;
case 101:
        taste = 101;
        break;
case 102:
        taste = 102;
        break;
case 160:
        taste = 160;
        break;
case 226:
        taste = 226;
        break;
case 89:
        taste = 89;
        break;
case 88:
        taste = 88;
        break;
case 67:
        taste = 67;
        break;
case 86:
        taste = 86;
        break;
case 66:
        taste = 66;
        break;
case 78:
        taste = 78;
        break;
case 77:
        taste = 77;
        break;
case 188:
        taste = 188;
        break;
case 190:
        taste = 190;
        break;
case 189:
        taste = 189;
        break;
case 161:
```

```

                taste = 161;
                break;
            case 38:
                taste = 38;
                break;
            case 97:
                taste = 97;
                break;

            case 98:
                taste = 98;
                break;
            case 99:
                taste = 99;
                break;
            case 162:
                taste = 162;
                break;
            case 91:
                taste = 91;
                break;
            case 164:
                taste = 164;
                break;
            case 32:
                taste = 32;
                break;
            case 165:
                taste = 165;
                break;

            case 93:
                taste = 93;
                break;
            case 163:
                taste = 163;
                break;
            case 37:
                taste = 37;
                break;
            case 40:
                taste = 40;
                break;
            case 39:
                taste = 39;
                break;
            case 96:
                taste = 96;
                break;
            case 110:
                taste = 110;
                break;

        }
    });
}

double[] h = new double[300];
double[] f = new double[300];
float[] g = new float[300];

```

```

int grosse = 65;
int abstandx = 70;
int a = abstandx / 2;
float c = (3 * abstandx) / 4;
float d = abstandx / 4;
int grossekreis = 75; //75

int i = grosse / 2;
int abstandy = 70;
int e = (1079 / 6 + 1079 / 5) / 2;
int b = abstandy / 2;
int durchsichtig = 25;

int color1 = color(0,0,128, 10); //noch testen
int color11 = color(0,0,128, 5);

int color2 = color(0,191,255, 10);
int color22 = color(0,191,255, 5);

int color3 = color(33,33,33, 10);
int color33 = color(33,33,33, 5);

static int taste;
Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-ss");

static boolean exit = false;

public void settings() {

    fullScreen();

}

public void setup() {

    beginRecord(PDF, "Programm5 Kyo Spiel " + ft.format(date) + ".pdf");
    background(255);
    smooth();
    noStroke();

}

public void draw() {

    switch (taste) {
    case 27:
        fill(color1);
        rect(abstandx, abstandy + e, grosse, grosse,
durchsichtig);

        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste])) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(abstandx + i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    }
}

```

```

case 112:

    fill(color1);
    rect(3 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

    fill(color11);
    ellipse(3 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
    f[taste]++;
    taste = 0;

    break;
case 113:
    fill(color1);
    rect(4 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

    fill(color11);
    ellipse(4 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
    f[taste]++;
    taste = 0;
    break;
case 114:
    fill(color1);
    rect(5 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

    fill(color11);
    ellipse(5 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
    f[taste]++;
    taste = 0;
    break;
case 115:
    fill(color1);
    rect(6 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

    fill(color11);
    ellipse(6 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
    f[taste]++;
    taste = 0;
    break;
case 116:
    fill(color1);
    rect(8 * abstandx - a, abstandy + e, grosse, grosse,
durchsichtig);

```

```

        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(8 * abstandx - a+ i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 117:
        fill(color1);
        rect(9 * abstandx - a, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(9 * abstandx- a + i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 118:
        fill(color1);
        rect(10 * abstandx - a, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(10 * abstandx - a+ i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 119:
        fill(color1);
        rect(11 * abstandx - a, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(11 * abstandx - a+ i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 120:
        fill(color1);
        rect(12 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(12 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);

```

```

        f[taste]++;
        taste = 0;
        break;
    case 121:
        fill(color1);
        rect(13 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(13 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 122:
        fill(color1);
        rect(14 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(14 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 123:
        fill(color1);
        rect(15 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(15 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 44:
        fill(color1);
        rect(17 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(17 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 145:
        fill(color1);
        rect(18 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);

```

```

        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(18 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 19:
        fill(color1);
        rect(19 * abstandx, abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(19 * abstandx + i, abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 144:
        fill(color2);
        rect(22 * abstandx + a, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(22 * abstandx + i+a, 3*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 111:
        fill(color2);
        rect(23 * abstandx + a, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(23 * abstandx + i+a, 3*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 106:
        fill(color2);
        rect(24 * abstandx + a, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(24 * abstandx + i+a, 3*abstandy + e + i,
g[taste], g[taste]);

```

```

        f[taste]++;
        taste = 0;
        break;
    case 109:
        fill(color2);
        rect(25 * abstandx + a, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(25 * abstandx + i+a, 3*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 49:
        fill(color3);
        rect(2 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(2 * abstandx + i, 3*abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 50:
        fill(color3);
        rect(3 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(3 * abstandx + i, 3*abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 51:
        fill(color3);
        rect(4 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(4 * abstandx + i, 3*abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 52:
        fill(color3);
        rect(5 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);

```

```

        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(5 * abstandx + i, 3*abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 53:
        fill(color3);
        rect(6 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(6 * abstandx + i, 3* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 54:
        fill(color3);
        rect(7 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(7 * abstandx + i, 3*abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 55:
        fill(color3);
        rect(8 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(8 * abstandx + i, 3*abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 56:
        fill(color3);
        rect(9 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(9 * abstandx + i, 3*abstandy + e + i, g[taste],
g[taste]);

```

```

        f[taste]++;
        taste = 0;
        break;
    case 57:
        fill(color3);
        rect(10 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(10 * abstandx + i, 3* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 48:
        fill(color3);
        rect(11 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(11 * abstandx + i, 3* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 219:
        fill(color3);
        rect(12 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(12 * abstandx + i, 3*abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 221:
        fill(color3);
        rect(13 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(13 * abstandx + i, 3* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 8:
        fill(color3);
        rect(14 * abstandx, 3 * abstandy + e, 2 * grosse, grosse,
durchsichtig);

```

```

        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(14 * abstandx + grosse ,3* abstandy + e + i,
g[taste]+grosse, g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 45:
        fill(color1);
        rect(17 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(17 * abstandx + i,3* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 36:
        fill(color1);
        rect(18 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(18 * abstandx + i,3* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 33:
        fill(color1);
        rect(19 * abstandx, 3 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(19 * abstandx + i,3* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 9:
        fill(color3);
        rect(abstandx, 4 * abstandy + e, grosse + a, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(abstandx + a/2+i,4* abstandy + e + i, g[taste]+a,
g[taste]);

```

```

        f[taste]++;
        taste = 0;
        break;
    case 81:
        fill(color3);
        rect(2 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(2 * abstandx+ a + i, 4*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 87:
        fill(color3);
        rect(3 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(3 * abstandx + a+ i, 4*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 69:
        fill(color3);
        rect(4 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(4 * abstandx + a+ i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 82:
        fill(color3);
        rect(5 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(5 * abstandx + a+ i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 84:
        fill(color3);
        rect(6 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);

```

```

        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(6 * abstandx + a+ i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 90:
        fill(color3);
        rect(7 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(7 * abstandx + a+ i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 85:
        fill(color3);
        rect(8 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(8 * abstandx + a+ i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 73:
        fill(color3);
        rect(9 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(9 * abstandx + a+ i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 79:
        fill(color3);
        rect(10 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(10 * abstandx+ a + i,4* abstandy + e + i,
g[taste], g[taste]);

```

```

        f[taste]++;
        taste = 0;
        break;
    case 80:
        fill(color3);
        rect(11 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(11 * abstandx+ a + i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 186:
        fill(color3);
        rect(12 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(12 * abstandx + a+ i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 192:
        fill(color3);
        rect(13 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(13 * abstandx+ a + i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 13:
        fill(color3);
        beginShape();
        vertex(14 * abstandx + a, 4 * abstandy + e);
        vertex(14 * abstandx + a + 90, 4 * abstandy + e);
        vertex(14 * abstandx + a + 90, 6 * abstandy - 5 + e);
        vertex(14 * abstandx + a + 20, 6 * abstandy - 5 + e);
        vertex(14 * abstandx + a + 20, 5 * abstandy + e);
        vertex(14 * abstandx + a, 5 * abstandy + e);
        endShape();
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(14 * abstandx + a+45,5* abstandy + e ,
g[taste]+90, g[taste]+2*abstandy);
        f[taste]++;

```

```

        taste = 0;
        break;
    case 46:
        fill(color1);
        rect(17 * abstandx, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(17 * abstandx + i, 4 * abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        ;
        break;
    case 35:
        fill(color1);
        rect(18 * abstandx, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(18 * abstandx + i, 4 * abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        ;
        break;
    case 34:
        fill(color1);
        rect(19 * abstandx, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(19 * abstandx + i, 4 * abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        ;
        break;
    case 103:
        fill(color2);
        rect(22 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(22 * abstandx + a + i, 4 * abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        ;
        break;
    case 104:

```

```

        fill(color2);
        rect(23 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(23 * abstandx+ a + i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 105:
        fill(color2);
        rect(24 * abstandx + a, 4 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(24 * abstandx + a+ i,4* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 107:
        fill(color2);
        rect(25 * abstandx + a, 4 * abstandy + e, grosse, 2 *
grosse, durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(25 * abstandx + a+ i,4* abstandy + e +
i+grosse/2, g[taste], g[taste]+grosse);
        f[taste]++;
        taste = 0;
        break;
    case 65:
        fill(color3);
        rect(2 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(2 * abstandx + c+ i,5* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 83:
        fill(color3);
        rect(3 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

```

```

        fill(color33);
        ellipse(3 * abstandx + c+ i,5* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 68:
        fill(color3);
        rect(4 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(4 * abstandx + c+ i,5* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 70:
        fill(color3);
        rect(5 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(5 * abstandx + c+ i, 5*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 71:
        fill(color3);
        rect(6 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(6 * abstandx + c+ i,5* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 72:
        fill(color3);
        rect(7 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(7 * abstandx + c+ i,5* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 74:

```

```

        fill(color3);
        rect(8 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(8 * abstandx + c+ i,5* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 75:
        fill(color3);
        rect(9 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(9 * abstandx + c+ i, 5*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 76:
        fill(color3);
        rect(10 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(10 * abstandx+ c + i,5* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 222:
        fill(color3);
        rect(11 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(11 * abstandx + c+ i,5* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 220:
        fill(color3);
        rect(12 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

```

```

        fill(color33);
        ellipse(12 * abstandx + c+ i, 5*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 223:
        fill(color3);
        rect(13 * abstandx + c, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(13 * abstandx + c+ i, 5*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 100:
        fill(color2);
        rect(22 * abstandx + a, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(22 * abstandx + a+ i, 5*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        ;
        taste = 0;
        break;
    case 101:
        fill(color2);
        rect(23 * abstandx + a, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(23 * abstandx + a+ i, 5*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 102:
        fill(color2);
        rect(24 * abstandx + a, 5 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(24 * abstandx + a+ i, 5*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;

```

```

case 160:
    fill(color3);
    rect(abstandx, 6 * abstandy + e, grosse + d, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

    fill(color33);
    ellipse(abstandx + i+d/2, 6* abstandy + e + i, g[taste]+d,
g[taste]);
    f[taste]++;
    taste = 0;
    break;
case 226:
    fill(color3);
    rect(2 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

    fill(color33);
    ellipse(2 * abstandx + d+ i, 6* abstandy + e + i,
g[taste], g[taste]);
    f[taste]++;
    taste = 0;
    break;
case 89:
    fill(color3);
    rect(3 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

    fill(color33);
    ellipse(3 * abstandx + d+ i, 6* abstandy + e + i,
g[taste], g[taste]);
    f[taste]++;
    taste = 0;
    break;
case 88:
    fill(color3);
    rect(4 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

    fill(color33);
    ellipse(4 * abstandx + d+ i, 6* abstandy + e + i,
g[taste], g[taste]);
    f[taste]++;
    taste = 0;
    break;
case 67:
    fill(color3);
    rect(5 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
    h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
    g[taste] = (float) h[taste];

```

```

        fill(color33);
        ellipse(5 * abstandx + d+ i,6* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 86:
        fill(color3);
        rect(6 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(6 * abstandx + d+ i,6* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 66:
        fill(color3);
        rect(7 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(7 * abstandx + d+ i, 6* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 78:
        fill(color3);
        rect(8 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(8 * abstandx + d+ i,6* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 77:
        fill(color3);
        rect(9 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(9 * abstandx + d+ i,6* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;

```

```

        case 188:
            fill(color3);
            rect(10 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(10 * abstandx + d+ i,6* abstandy + e + i,
g[taste], g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 190:
            fill(color3);
            rect(11 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(11 * abstandx + d+ i,6* abstandy + e + i,
g[taste], g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 189:
            fill(color3);
            rect(12 * abstandx + d, 6 * abstandy + e, grosse, grosse,
durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(12 * abstandx + d+ i,6* abstandy + e + i,
g[taste], g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 161:
            fill(color3);
            rect(13 * abstandx + d, 6 * abstandy + e, 3 * grosse -
10, grosse, durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(13 * abstandx +d+(3 * grosse - 10)/2,6* abstandy
+ e + i, g[taste]+(3 * grosse - 10), g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 38:
            fill(color1);
            rect(18 * abstandx, 6 * abstandy + e, grosse, grosse,
durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

```

```

        fill(color11);
        ellipse(18 * abstandx + i, 6* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 97:
        fill(color2);
        rect(22 * abstandx + a, 6 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(22 * abstandx + a+ i, 6* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 98:
        fill(color2);
        rect(23 * abstandx + a, 6 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(23 * abstandx+ a + i, 6* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 99:
        fill(color2);
        rect(24 * abstandx + a, 6 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color22);
        ellipse(24 * abstandx + a+ i, 6* abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 162:
        fill(color3);
        rect(abstandx, 7 * abstandy + e, grosse + a, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(abstandx + i+a/2, 7*abstandy + e + i, g[taste]+a,
g[taste]);
        f[taste]++;
        taste = 0;
        break;

```

```

        case 91:
            fill(color3);
            rect(2 * abstandx + a, 7 * abstandy + e, grosse, grosse,
durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(2 * abstandx + a+ i,7* abstandy + e + i,
g[taste], g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 164:
            fill(color3);
            rect(3 * abstandx + a, 7 * abstandy + e, grosse + d,
grosse, durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(3 * abstandx + a+ i+d/2,7* abstandy + e + i,
g[taste]+d, g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 32:
            fill(color3);
            rect(4 * abstandx + c, 7 * abstandy + e, 6 * grosse + c +
7, grosse, durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(4 * abstandx + c+(6*grosse+c+7)/2,7* abstandy + e
+ i, g[taste]+c+7+6*grosse, g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 165:
            fill(color3);
            rect(11 * abstandx + d, 7 * abstandy + e, grosse + d,
grosse, durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(11 * abstandx + d+ i+d/2,7* abstandy + e + i,
g[taste]+d, g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 93:
            fill(color3);
            rect(13 * abstandx + a, 7 * abstandy + e, grosse, grosse,
durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];
    
```

```

        fill(color33);
        ellipse(13 * abstandx + a+ i, 7*abstandy + e + i,
g[taste], g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 163:
        fill(color3);
        rect(14 * abstandx + a, 7 * abstandy + e, grosse + a,
grosse, durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color33);
        ellipse(14 * abstandx + a+ i+a/2,7* abstandy + e + i,
g[taste]+a, g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 37:
        fill(color1);
        rect(17 * abstandx, 7 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(17 * abstandx + i,7* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 40:
        fill(color1);
        rect(18 * abstandx, 7 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(18 * abstandx + i, 7*abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;
    case 39:
        fill(color1);
        rect(19 * abstandx, 7 * abstandy + e, grosse, grosse,
durchsichtig);
        h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
        g[taste] = (float) h[taste];

        fill(color11);
        ellipse(19 * abstandx + i,7* abstandy + e + i, g[taste],
g[taste]);
        f[taste]++;
        taste = 0;
        break;

```

```

        case 96:
            fill(color2);
            rect(22 * abstandx + a, 7 * abstandy + e, 2 * grosse + 5,
grosse, durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color22);
            ellipse(22 * abstandx + a+ i+(grosse+5)/2,7* abstandy + e
+ i, g[taste]+grosse+5, g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 110:
            fill(color2);
            rect(24 * abstandx + a, 7 * abstandy + e, grosse, grosse,
durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color22);
            ellipse(24 * abstandx+ a + i,7* abstandy + e + i,
g[taste], g[taste]);
            f[taste]++;
            taste = 0;
            break;
        case 20:
            fill(color3);
            rect(abstandx, 5 * abstandy + e, grosse + c, grosse,
durchsichtig);
            h[taste] = (grossekreis * -Math.exp(-0.1 * f[taste]) +
grossekreis) * Math.pow(f[taste], 0.333);
            g[taste] = (float) h[taste];

            fill(color33);
            ellipse(abstandx + c/2+i, 5*abstandy + e + i, g[taste]+c,
g[taste]);
            f[taste]++;
            taste = 0;
            break;
    }

    if (exit == true) {
        endRecord();
        saveFrame("Programm5 Kyo Spiel " + ft.format(date) + "
.tif");
        exit();
    }
}

```

9.2.6 Programm 6

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;

```

```

import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import javax.swing.*;

public class Programm6stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(42000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm6stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;
                        break;
                }
            }
        });
    }
}

```

```

        // System.out.println(event);

    }

    @Override
    public void keyReleased(GlobalKeyEvent event) {
        // System.out.println(event);

    }
});

}

static int x, y;

int i;

int a = 0;
int b = 0;
int c = 128;

int f = 0;
int g = 191;
int h = 255;

int l = 33;
int m = 33;
int o = 33;

int j = 100;
int color1 = color(f, g, h, j);
int color2 = color(a, b, c, j);
int color3 = color(l, m, o, j);
Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss ");
static boolean exit = false;

public void settings() {

    fullScreen();

}

public void setup() {

    beginRecord(PDF, "Programm6 Kyo Spiel " + ft.format(date) + "
.pdf");
    background(255);
    smooth();
    strokeCap(PROJECT);
    strokeWeight(3);
}

public void draw() {
    float x = MouseInfo.getPointerInfo().getLocation().x; /* always
1 */
    float y = MouseInfo.getPointerInfo().getLocation().y; /* always
1 */

    if (x == 1919 || x == 0) {/* Ränder */

```

```

        if (i > 0) {
            noStroke();
            fill(color1);
            ellipse(x, y, 100, 100);

            stroke(color1);

            line(x, y, 1919 / 2, 1079 / 2);

            i = -10;
        }
    }
    if (y == 1079) {
        if (i > 0) {
            noStroke();
            fill(color2);
            ellipse(x, y, 100, 100);

            stroke(color2);

            line(x, y, 1919 / 2, 1079 / 2);

            i = -10;
        }
    }
    if (y == 0) {
        if (i > 0) {
            noStroke();
            fill(color3);
            ellipse(x, y, 100, 100);

            stroke(color3);

            line(x, y, 1919 / 2, 1079 / 2);

            i = -10;
        }
    }
}

if (exit == true) {
    endRecord();
    saveFrame("Programm6 Kyo Spiel " + ft.format(date) + ".tif");
    exit();
}

i++;
}
}

```

9.2.7 Programm 7

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;

```

```

import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import javax.swing.*;

public class Programm7stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(40000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm7stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;
                        break;
                }
                // System.out.println(event);
            }
        });
    }
}

```

```

    });
    GlobalMouseHook mouseHook = new GlobalMouseHook();
    mouseHook.addMouseListener(new GlobalMouseAdapter() {

        @Override
        public void mouseWheel(GlobalMouseEvent event) {
            // System.out.println(event);
            // size=size+1;
            wheel=true;

        }
    });

    static double size = 2;
    int zahler;
    static boolean wheel = false;
    int mode = 2;

    int ab = 0;
    int ac = 0;
    int ad = 128;

    int ae = 0;
    int af = 191;
    int ag = 255;

    int ah = 33;
    int ai = 33;
    int aj = 33;

    int ak = ae;
    int al = af;
    int am = ag;
    float start1 = 1;
    float start2;
    float xPos;
    float yPos = height / 2;
    double u = -0.001;

    float rad = 75;
    Date date = new Date();
    SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss ");
    static boolean exit = false;

    public void settings() {

        fullScreen();
    }

    public void setup() {

        beginRecord(PDF, "Programm7 Kyo Spiel " + ft.format(date) + "
.pdf");
        background(255);
        smooth();

        fill(0);
        // noFill();
    }
}

```

```

//strokeWeight(100);
}

public void draw() {
    if(wheel==false&&size>2) {
        zahler++;
    }
    if (wheel==true){
        size += 1;
        wheel=false;
        zahler=0;
    }
    if(zahler>20&&size>2) {
        size-=0.1;
    }

    // fill(0,20);
    // rect(0,0,width,height);

    // float x = MouseInfo.getPointerInfo().getLocation().x; /*
always 1 */
    // float y = MouseInfo.getPointerInfo().getLocation().y; /*
always 1 */

    if (xPos > width) {
        xPos = 0;
    }
    // xPos=(int) (noise(start1+=u)*height+1);
    xPos = xPos + 1;
    yPos = (int) (noise(start2 += u) * height);
    if (mode == 1) {
        if (ak < ab && ak != ab) {
            ak++;
        } else if (ak != 0) {
            ak--;
        }

        if (al < ac && al != ac) {
            al++;
        } else if (al != 0) {
            al--;
        }
        if (am < ad && am != ad) {
            am++;
        } else if (am != 0) {
            am--;
        }
        if ((ak == ab || ak == ab + 1 || ak == ab - 1) && (al ==
ac || al == ac - 1 || al == ac + 1)
                && (am == ad || am == ad + 1 || am == ad - 1))
    {
        mode = 2;
    }
}
if (mode == 2) {
    if (ak < ae) {
        ak++;
    } else if (ak != 0) {
        ak--;
    }

    if (al < af) {
        al++;
    }
}

```

```

        } else if (al != 0) {
            al--;
        }
        if (am < ag) {
            am++;
        } else if (am != 0) {
            am--;
        }
        if ((ak == ae || ak == ae + 1 || ak == ae - 1) && (al ==
af || al == af - 1 || al == af + 1)
                && (am == ag || am == ag + 1 || am == ag - 1))
{
            mode = 3;
        }

    }
    if (mode == 3) {
        if (ak < ah) {
            ak++;
        } else if (ak != 0) {
            ak--;
        }

        if (al < ai) {
            al++;
        } else if (al != 0) {
            al--;
        }
        if (am < aj) {
            am++;
        } else if (am != 0) {
            am--;
        }
        if ((ak == ah || ak == ah + 1 || ak == ah - 1) && (al ==
ai || al == ai - 1 || al == ai + 1)
                && (am == aj || am == aj + 1 || am == aj - 1))
{
            mode = 1;
        }
    }
    stroke(ak, al, am); //testen
    float size2 = (float) size;
    double c = (50 * -Math.exp(-0.1 * size) + 50) * Math.pow(size,
0.333);

    float g = (float) c;
    ellipse(xPos, yPos, g, g);

    // ellipse(xPos, yPos, size2, size2);

    if (exit == true) {
        endRecord();
        saveFrame("Programm7 Kyo Spiel " + ft.format(date) + "
.tif");
        exit();
    }
}

```

9.2.8 Programm 8

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import javax.swing.*;

public class Programm8stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(38000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm8stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

```

```

        switch (event.getVirtualKeyCode()) {

    case 191:
        exit = true;
        break;
    }

    // System.out.println(event);

}

@Override
public void keyReleased(GlobalKeyEvent event) {
    // System.out.println(event);

}
});

GlobalMouseHook mouseHook = new GlobalMouseHook();
mouseHook.addMouseListener(new GlobalMouseAdapter() {
    @Override
    public void mousePressed(GlobalMouseEvent event) {
        // System.out.println(event);
        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_LEFT) != GlobalMouseEvent.BUTTON_NO) {
            button = 1;

            i++;
            k = 0;
            float x = MouseInfo.getPointerInfo()
                .getLocation().x; /* always 1 */
            float y = MouseInfo.getPointerInfo()
                .getLocation().y; /* always 1 */

            e = x;
            d = y;

        }

        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_RIGHT) != GlobalMouseEvent.BUTTON_NO) {
            button = 2;

            i++;
            k = 0;
            float x = MouseInfo.getPointerInfo()
                .getLocation().x; /* always 1 */
            float y = MouseInfo.getPointerInfo()
                .getLocation().y; /* always 1 */

            e = x;
            d = y;
            // println("rechts");
        }

        if (event.getButton() ==
GlobalMouseEvent.BUTTON_MIDDLE) {
            button = 3;

            i++;
            k = 0;
            float x = MouseInfo.getPointerInfo()
                .getLocation().x; /* always 1 */

```

```

        float y = MouseInfo.getPointerInfo()
            .getLocation().y; /* always 1 */

        e = x;
        d = y;
        // println("middle");
    }

}

@Override
public void mouseReleased(GlobalMouseEvent event) {

    // System.out.println(event);

    j++;
    b = MouseInfo.getPointerInfo()
        .getLocation().x; /* always last */
    c = MouseInfo.getPointerInfo()
        .getLocation().y; /* always last */

}

});

}

static int button;
static boolean makeline = false;
static float e, d, q, w, r, t, z;
static boolean mousemove = false;
static int mouseclick = 0;
static boolean mouserelease = false;
static int i, j, k;
static int x, y;
static int b, c;

int ab = 0;
int ac = 0;
int ad = 128;

int ae = 0;
int af = 191;
int ag = 255;

int ah = 33;
int ai = 33;
int aj = 33;

int color1 = color(ab, ac, ad, 150);
int color2 = color(ae, af, ag, 150);
int color3 = color(ah, ai, aj, 150);

Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-ss");

static boolean exit = false;

public void settings() {

    fullScreen();
}

```

```

}

public void setup() {
    beginRecord(PDF, "Programm8 Kyo Spiel " + ft.format(date) + "
.pdf");
    background(255);
    smooth();

    strokeWeight(30);

    strokeCap(SQUARE);
}
public void draw() {

    if((i-j)>=2) {
        button=0;
        println("hi");
        i=j;
    }
    if((j-i)>=2) {
        button=0;
        println("hi");
        i=j;
    }
    switch (button) {
    case 1:
        stroke(color1);
        float ff = dist(e, d, b, c);
        if (ff > 5 && i == j && k == 0) {

            line(e, d, b, c);
            k = 1;
        }
        break;
    case 2:
        stroke(color2);
        ff = dist(e, d, b, c);
        if (ff > 5 && i == j && k == 0) {

            line(e, d, b, c);
            k = 1;
        }
        break;
    case 3:
        stroke(color3);
        ff = dist(e, d, b, c);
        if (ff > 5 && i == j && k == 0) {

            line(e, d, b, c);
            k = 1;
        }
        break;
    }

    if (exit == true) {
        endRecord();
        saveFrame("Programm8 Kyo Spiel " + ft.format(date) + "
.tif");
        exit();
    }
}
}

```

9.2.9 Programm 9

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Programm9stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(36000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm9stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

```

```

        i++;
        switch (event.getVirtualKeyCode()) {

    case 191:
        exit = true;
        break;
    }

    // System.out.println(event);

}

@Override
public void keyReleased(GlobalKeyEvent event) {

    k++;
    // System.out.println(event);

}
});

GlobalMouseHook mouseHook = new GlobalMouseHook();
mouseHook.addMouseListener(new GlobalMouseAdapter() {
    @Override
    public void mousePressed(GlobalMouseEvent event) {
        // System.out.println(event);
        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_LEFT) != GlobalMouseEvent.BUTTON_NO) {
            o++;
            // println("links");

        }

        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_RIGHT) != GlobalMouseEvent.BUTTON_NO) {
            m++;
            // println("rechts");
        }

        if (event.getButton() ==
GlobalMouseEvent.BUTTON_MIDDLE) {
            n++;
            // println("middle");
        }
    }
});
}

int ab = 0;
int ac = 0;
int ad = 128;

int ae = 0;
int af = 191;
int ag = 255;

int ah = 33;
int ai = 33;
int aj = 33;

```

```

int color1 = color(ab, ac, ad, 122);
int color2 = color(ae, af, ag, 122);
int color3 = color(ah, ai, aj, 122);
int mode = 1;
static int y = 15;
static int x = 15;
static int abstand=5;
static int i = -1;
static int j = 0;
static int k = -1;
static int m = -1;
static int n = 0;
static int o = -1;
static int l = 1080 / (y + abstand) - 1;
int p = 1920 / (x + abstand) - 1;
// int p = 319;

static boolean press = false;
static boolean release = false;

static float a, b;

Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss");

static boolean exit = false;

public void settings() {
    fullScreen();
}

public void setup() {
    beginRecord(PDF, "Programm9 Kyo Spiel " + ft.format(date) + ".pdf");
    background(255);
    smooth();
    noStroke();
}

public void draw() {
    if (exit == true) {
        for (a = 0; a < i; a++) { //i
            fill(color1);

            rect((x + abstand) * b, (y + abstand) * j, x, y);
            if ((x + abstand) * b > 1919) {
                b = -1;
                j++;
            }
            b++;
        }
        b = 0;
        for (a = 0; a < k; a++) { //k
            fill(color2);
        }
    }
}

```

```

rect((x + abstand) * b, (y + abstand) * l, x, y);
if ((x + abstand) * b > 1919) {
    b = -1;
    l--;
}
b++;

}

b = 0;
j = 0;
for (a = 0; a < o; a++) { //o
    fill(color3);

    rect((x + abstand) * j, (y + abstand) * b, x, y);
    if ((y + abstand) * b > 1079) {
        j++;
        b = -1;
    }
    b++;
}
b = 0;
j = 0;

for (a = 0; a < m; a++) { //m
    fill(color3);

    rect((x + abstand) * p, (y + abstand) * j, x, y);
    if ((y + abstand) * j > 1079) {
        j = -1;
        p--;
    }
    j++;
}

}

fill(255, 100);
ellipse(width / 2, height / 2, 50 * n, 50 * n);
ellipse(width / 10, height / 10*9, 6 * n, 6 * n);
ellipse(width / 10*9, height / 10*9, 6 * n, 6 * n);
ellipse(width / 10, height / 10, 6 * n, 6 * n);
ellipse(width / 10*9, height / 10, 6 * n, 6 * n);

endRecord();
saveFrame("Programm9 Kyo Spiel " + ft.format(date) + "
.tif");
exit();
}
}

```

9.2.10 Programm 10

```

import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Programm10stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(34000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm10stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

```

```

        switch (event.getVirtualKeyCode()) {

    case 191:
        exit = true;
        break;
    }

    // System.out.println(event);

}

@Override
public void keyReleased(GlobalKeyEvent event) {
    mode = 1;

    ar=true;
    i++;
    // System.out.println(event);

}
});

GlobalMouseHook mouseHook = new GlobalMouseHook();
mouseHook.addMouseListener(new GlobalMouseAdapter() {
    @Override
    public void mousePressed(GlobalMouseEvent event) {
        // System.out.println(event);
        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_LEFT) != GlobalMouseEvent.BUTTON_NO) {
            mode = 2;

            ar=true;
            i++;
            // println("links");

        }

        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_RIGHT) != GlobalMouseEvent.BUTTON_NO) {
            mode = 3;

            ar=true;
            i++;
            // println("rechts");
        }

        if (event.getButton() ==
GlobalMouseEvent.BUTTON_MIDDLE) {
            n++;
            // println("middle");
        }
    }
});

}

int ab = 0;
int ac = 0;
int ad = 128;

```

```

int ae = 0;
int af = 191;
int ag = 255;

int ah = 33;
int ai = 33;
int aj = 33;

int color1 = color(ab, ac, ad);
int color2 = color(ae, af, ag);
int color3 = color(ah, ai, aj);

static boolean ar=false;
static int mode = 1;

static int i=0;
int j=1;
static int n=0;

Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-ss");

static boolean exit = false;

public void settings() {

    fullScreen();

}

public void setup() {

    beginRecord(PDF, "Programm10 Kyo Spiel " + ft.format(date) + ".pdf");
    background(255);
    smooth();
    rectMode(CENTER);
    noStroke();

}

public void draw() {
    switch (mode){
        case 1 : fill(color1);break;
        case 2 : fill(color2);break;
        case 3 : fill(color3);break;
    }
    translate(width/2,height/2);

    if (ar==true&&j==i) {
        rotate((float)(0.1)*i);
        ellipse(i,0,10,10);
        ar=false;
        j++;
    }
    if (i>j){
        i=j;
    }
}

```

```
}

if (exit == true) {
    fill(0, 30);
    for(int k=0;k<10*n;k++) {
        rotate((float)-0.1);
        ellipse(k,0,10,10);
    }
    //    ellipse(0, 0, 50 * n, 50 * n);

    endRecord();
    saveFrame("Programm10_Kyo_Spiel " + ft.format(date) + ".tif");
    exit();
}

}
```

9.2.11 Programm 11

```
import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;
import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
```

```

import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Programm11stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(32000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm11stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;
                        break;
                }

                // System.out.println(event);

            }

            @Override
            public void keyReleased(GlobalKeyEvent event) {
                // System.out.println(event);

            }
        });

        GlobalMouseHook mouseHook = new GlobalMouseHook();
        mouseHook.addMouseListener(new GlobalMouseAdapter() {
            @Override
            public void mousePressed(GlobalMouseEvent event) {
                // System.out.println(event);
                if ((event.getButtons() &
GlobalMouseEvent.BUTTON_LEFT) != GlobalMouseEvent.BUTTON_NO) {
                    mode = 1;
                    f++;
                    // println("links");
                }

                if ((event.getButtons() &
GlobalMouseEvent.BUTTON_RIGHT) != GlobalMouseEvent.BUTTON_NO) {
                    mode = 2;
                }
            }
        });
    }
}

```

```

        f++;
        // println("rechts");
    }

    if (event.getButton() ==
GlobalMouseEvent.BUTTON_MIDDLE) {
        mode=3;
        f++;
        // println("middle");
    }

}

@Override
public void mouseReleased(GlobalMouseEvent event) {
    g++;
    // System.out.println(event);
}
);

}

static int mode;

int ab = 0;
int ac = 0;
int ad = 128;

int ae = 0;
int af = 191;
int ag = 255;

int ah = 33;
int ai = 33;
int aj = 33;

int color1 = color(ab, ac, ad,10);
int color2 = color(ae, af, ag,10);
int color3 = color(ah, ai, aj,10);

int[] q = new int[1000000];
int[] w = new int[1000000];
int[] e = new int[1000000];
int[] r = new int[1000000];
int i = 0;
int a, b;
static int f, g;
Date date = new Date();
SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss ");

static boolean exit = false;

public void settings() {

    fullScreen();
}

public void setup() {

```

```

beginRecord(PDF, "Programm11 Kyo Spiel " + ft.format(date) + "
.pdf");
background(255);
smooth();
stroke(0, 10);
// frameRate(240);
}

public void draw() {
    switch (mode) {
        case 1:
            stroke(color1);break;
        case 2:
            stroke(color2);break;
        case 3:
            stroke(color3);break;
    }
    if (f > g) {

        int x = MouseInfo.getPointerInfo().getLocation().x; /* always 1 */
        int y = MouseInfo.getPointerInfo().getLocation().y; /* always 1 */
        if (a != x || b != y) {
            for (int g = 0; g < i; g++) {
                if (dist(x, y, e[g], r[g]) < 100) {

                    line(x, y, e[g], r[g]);
                }
            }
            e[i] = x;
            r[i] = y;
            ia = x;
            b = y;
        }
        if (exit == true) {
            endRecord();
            saveFrame("Programm11 Kyo Spiel " + ft.format(date) + "
.tif");
            exit();
        }
    }
}

```

9.2.12 Programm 12

```
import processing.core.PApplet;
import java.awt.Frame;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import lc.kra.system.mouse.GlobalMouseHook;
import lc.kra.system.mouse.event.GlobalMouseAdapter;
import lc.kra.system.mouse.event.GlobalMouseEvent;
import processing.pdf.*;
import java.util.*;
```

```

import java.awt.*;
import java.awt.event.*;
import java.awt.MouseInfo;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.text.SimpleDateFormat;

import org.jnativehook.GlobalScreen;
import org.jnativehook.NativeHookException;
import org.jnativehook.keyboard.NativeKeyEvent;
import org.jnativehook.keyboard.NativeKeyListener;
import javax.swing.*;
import javax.swing.JFrame;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import lc.kra.system.keyboard.GlobalKeyboardHook;
import lc.kra.system.keyboard.event.GlobalKeyAdapter;
import lc.kra.system.keyboard.event.GlobalKeyEvent;
import javax.swing.JPanel;
import processing.core.*;
import processing.core.PApplet;
import java.awt.*;
import java.awt.event.KeyListener;
import java.awt.event.KeyListener;
import processing.pdf.*;
import java.awt.GridLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.eventMouseListener;
import java.awt.event.MouseEvent;

import javax.swing.*;

public class Programm12stable extends PApplet

{
    public static void main(String[] args) {
        try {
            Thread.sleep(30000);
        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        String[] a = { "MAIN" };
        PApplet.runSketch(a, new Programm12stable());
        GlobalKeyboardHook keyboardHook = new GlobalKeyboardHook();
        keyboardHook.addKeyListener(new GlobalKeyAdapter() {
            @Override
            public void keyPressed(GlobalKeyEvent event) {

                switch (event.getVirtualKeyCode()) {

                    case 191:
                        exit = true;
                        break;
                }
                // System.out.println(event);
            }
        });
    }
}

```

```

@Override
public void keyReleased(GlobalKeyEvent event) {
    // System.out.println(event);

}

});

GlobalMouseHook mouseHook = new GlobalMouseHook();
mouseHook.addMouseListener(new GlobalMouseAdapter() {
    @Override
    public void mousePressed(GlobalMouseEvent event) {
        // System.out.println(event);
        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_LEFT) != GlobalMouseEvent.BUTTON_NO) {
            mouse = true;

            // println("links");

        }

        if ((event.getButtons() &
GlobalMouseEvent.BUTTON_RIGHT) != GlobalMouseEvent.BUTTON_NO) {
            mouse = true;

            // println("rechts");
        }

        if (event.getButton() ==
GlobalMouseEvent.BUTTON_MIDDLE) {
            mouse = true;

            // println("middle");
        }

    }

    @Override
    public void mouseReleased(GlobalMouseEvent event) {
        // System.out.println(event);
    }

    @Override
    public void mouseMoved(GlobalMouseEvent event) {

        // System.out.println(event);
    }

    @Override
    public void mouseWheel(GlobalMouseEvent event) {
        // System.out.println(event);

    }
});

}

int mode =0;
float q = 0;
float rad = 75;
static int x, y;
static float b,c,d,e,k;
static boolean mouse = false;
Date date = new Date();

```

```

SimpleDateFormat ft = new SimpleDateFormat("E dd.MM.yyyy 'at' HH-mm-
ss ");
static boolean exit = false;

public void settings() {
    fullScreen();
}

public void setup() {
    beginRecord(PDF, "Programm12 Kyo Spiel " + ft.format(date) + ".pdf");
    background(255);
    smooth();
    stroke(0,200);
}

public void draw() {

    float x = MouseInfo.getPointerInfo().getLocation().x; /* always
1 */
    float y = MouseInfo.getPointerInfo().getLocation().y; /* always
1 */
    float xcenter = x;
    float ycenter = y;
    float angle;

    if (mouse == true) {
        if (x==b&&y==c) {
            for (float i = 0; i < 360; i += 1) {
                rad = random (0,100);
                float q = xcenter + cos(i) * rad;
                float w = ycenter + sin(i) * rad;
                rad = random (5,50);
                x = xcenter + cos(i) *rad ;
                y = ycenter + sin(i) * rad;

                line(x, y, q, w);
            }
            mouse = false;
        }
        b=x;
        c=y;
    }

    if (exit == true) {
        endRecord();
        saveFrame("Programm12 Kyo Spiel " + ft.format(date) + ".tif");
        exit();
    }
}

```

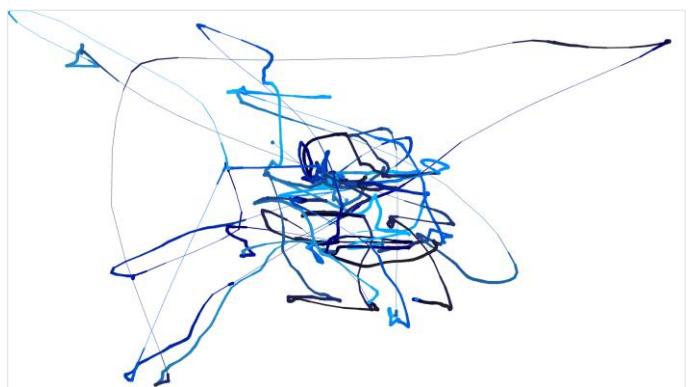
9.3 Bilder

9.3.1 Programm 1

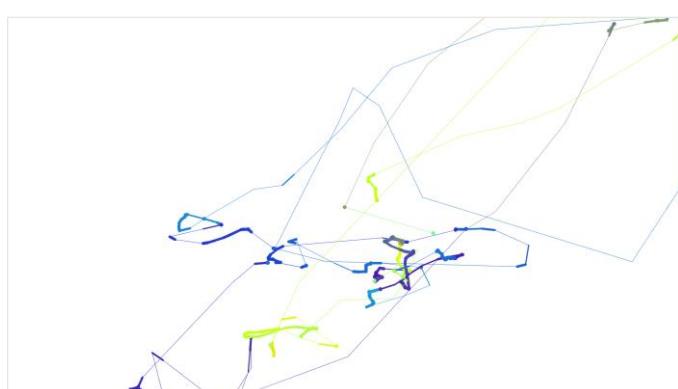
Aufsatz



Fiona



Kyo



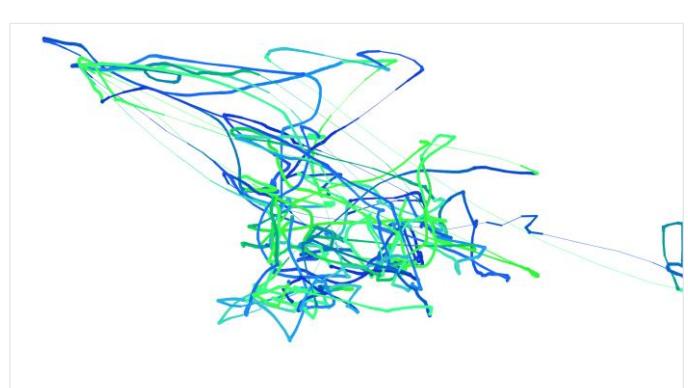
Gabriela



Simon



Julia

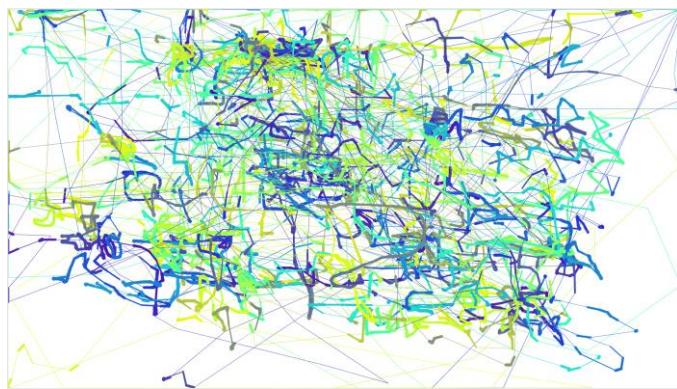


Andi

Bild



Fiona



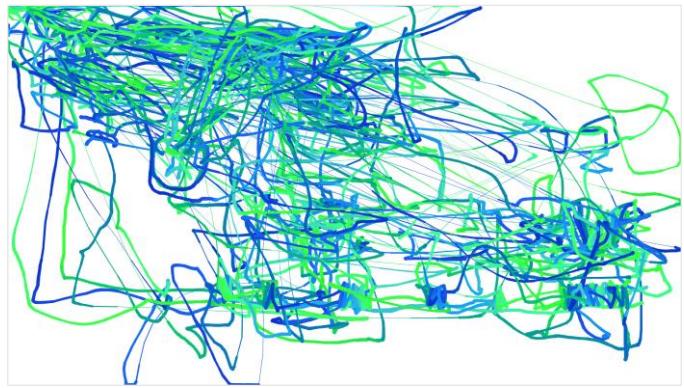
Kyo



Gabriela

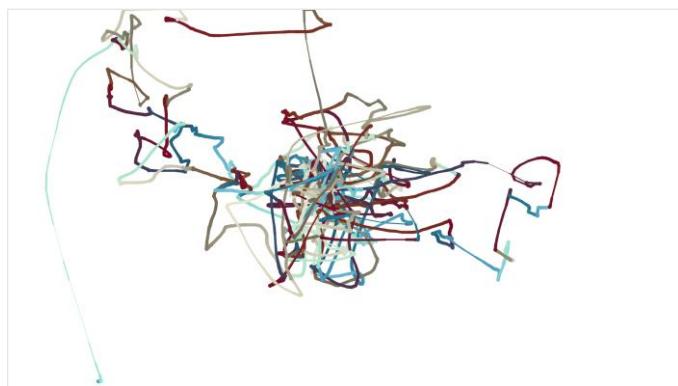


Simon



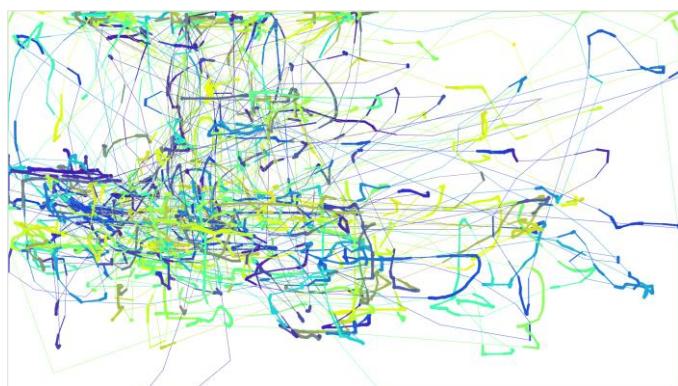
Julia

Andi



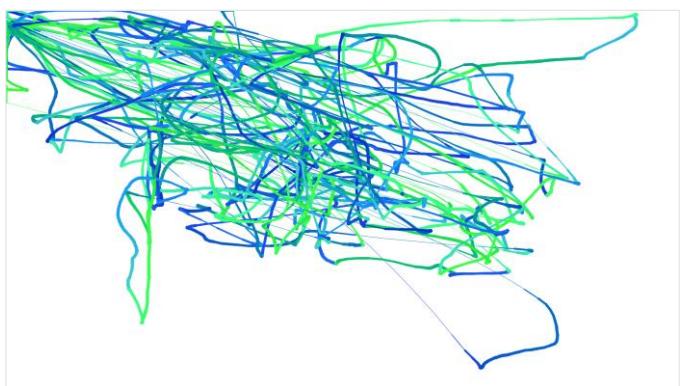
Fiona

Kyo



Gabriela

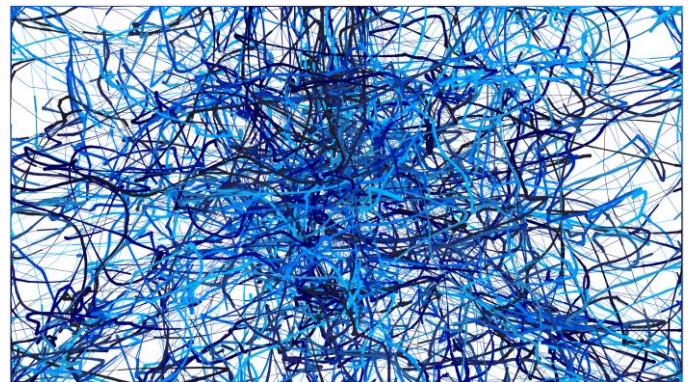
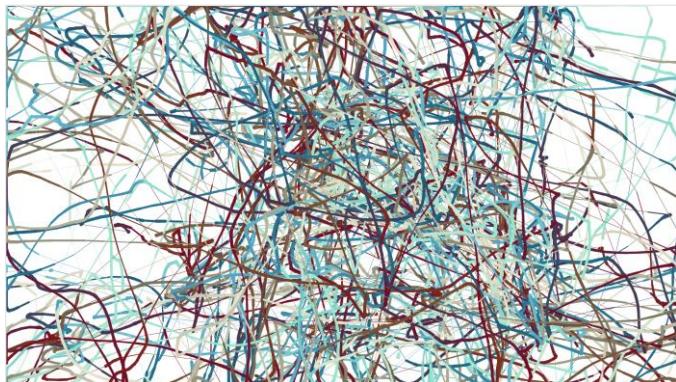
Simon



Julia

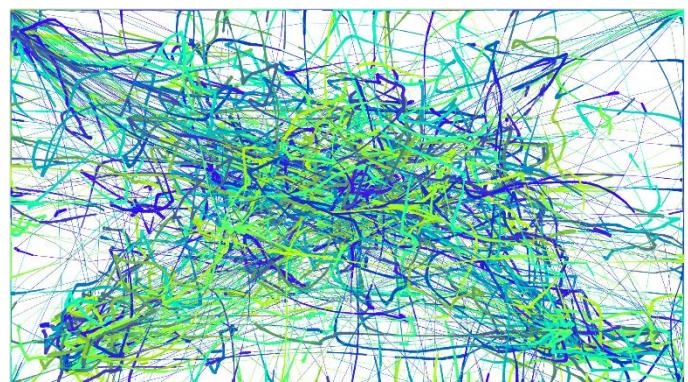
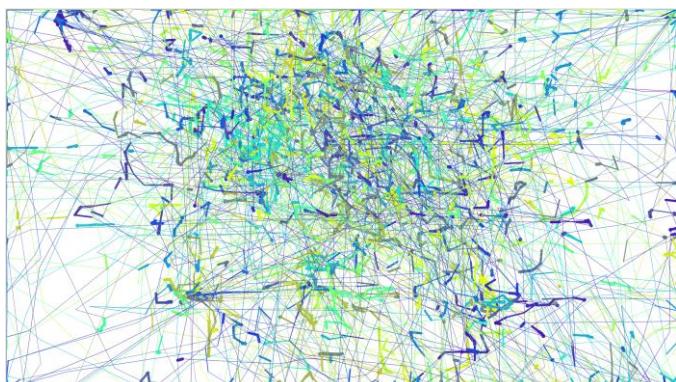
Andi

Spiel



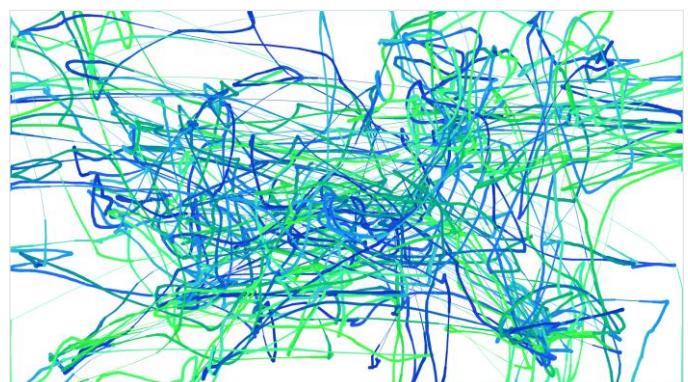
Fiona

Kyo



Gabriela

Simon

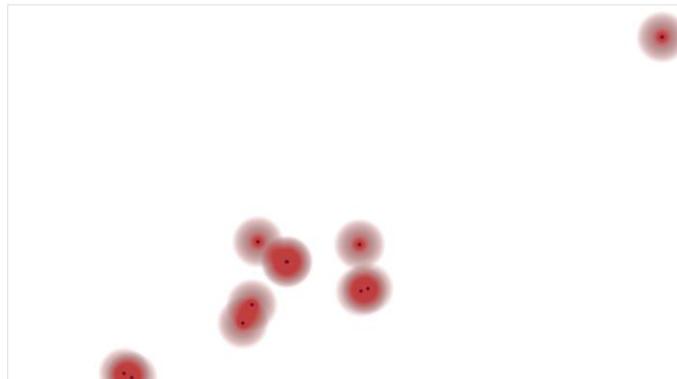


Julia

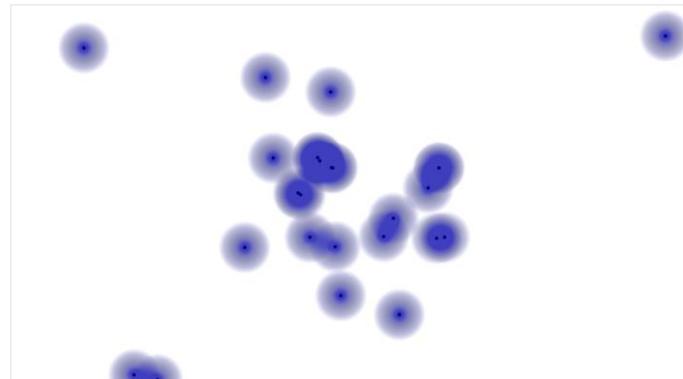
Andi

9.3.2 Programm 2

Aufsatz



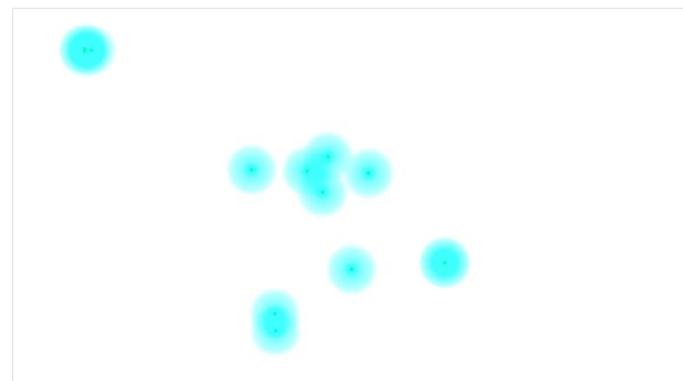
Fiona



Kyo



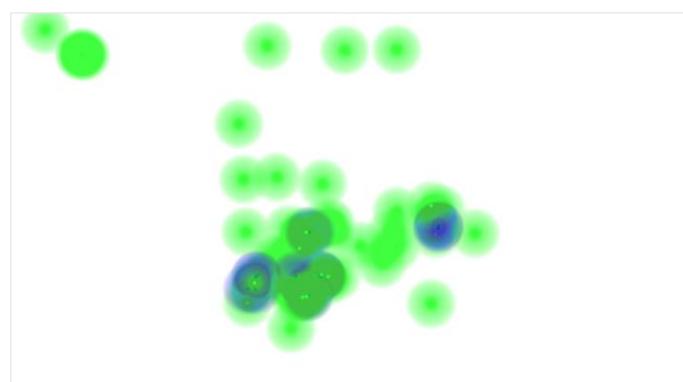
Gabriela



Simon

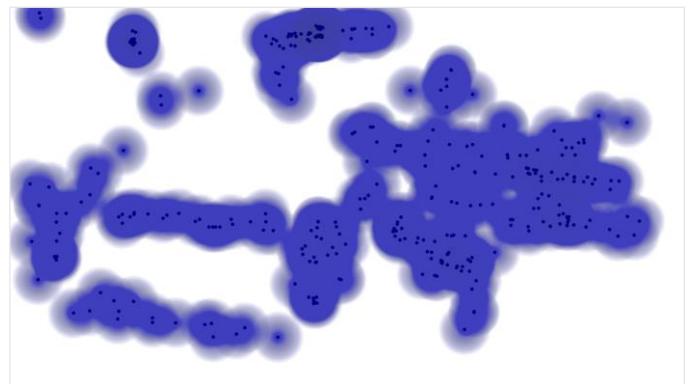
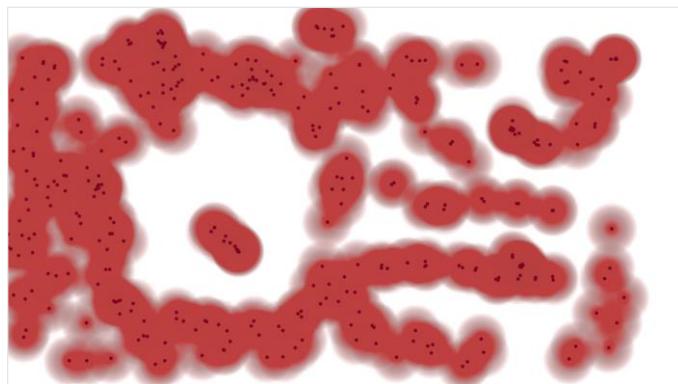


Julia

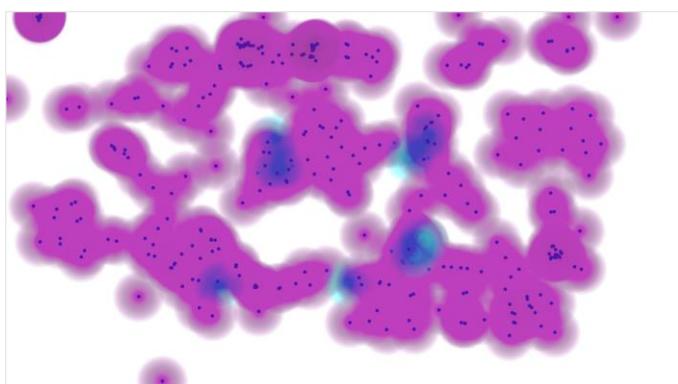


Andi

Bild



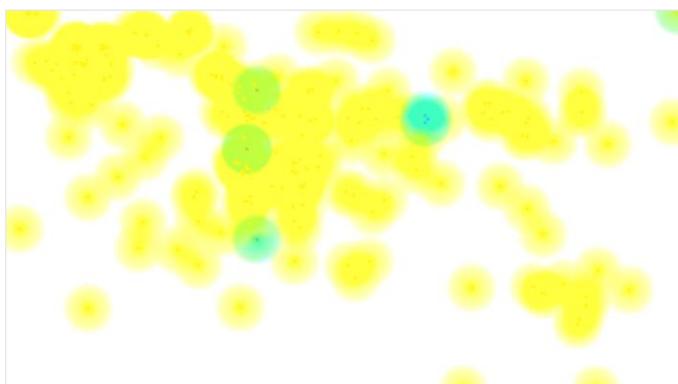
Fiona



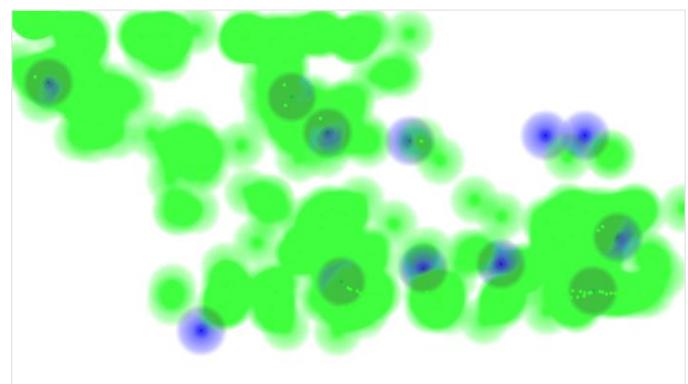
Kyo



Gabriela

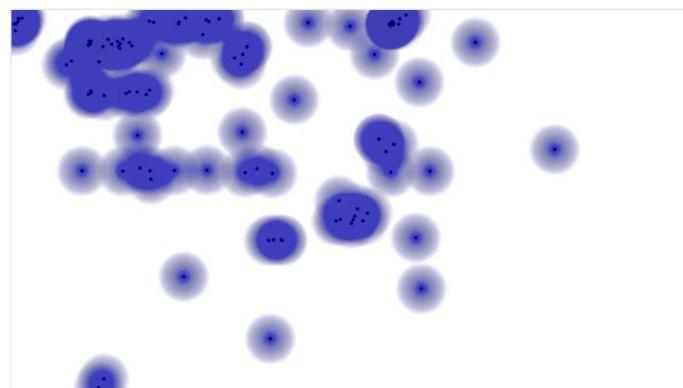


Simon

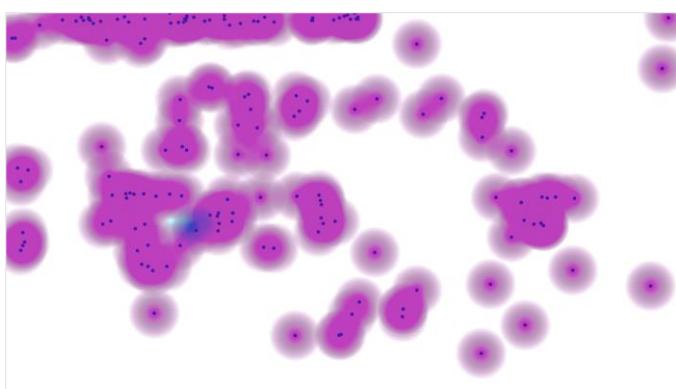


Julia

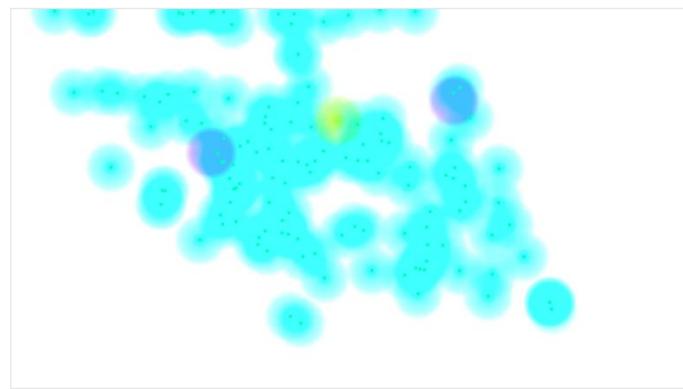
Andi



Fiona



Kyo



Gabriela

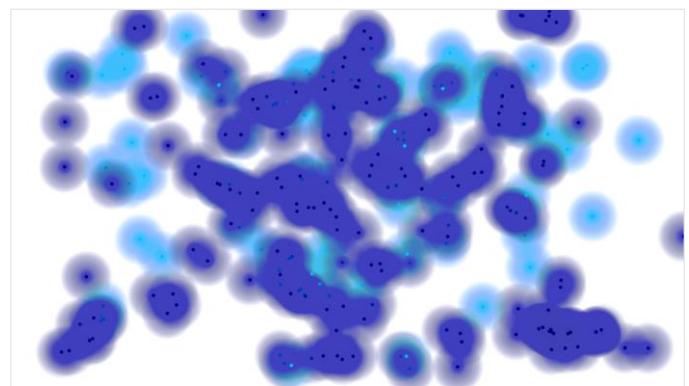
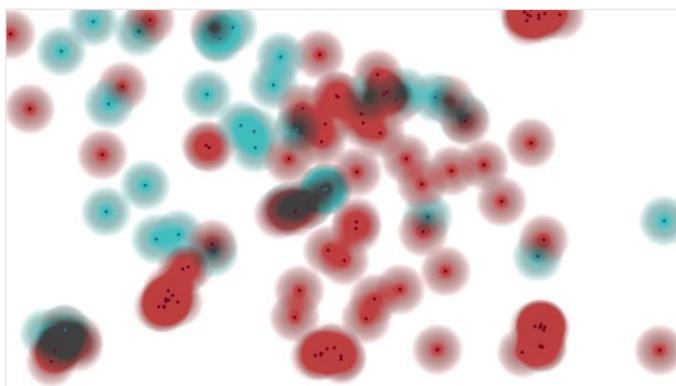


Simon

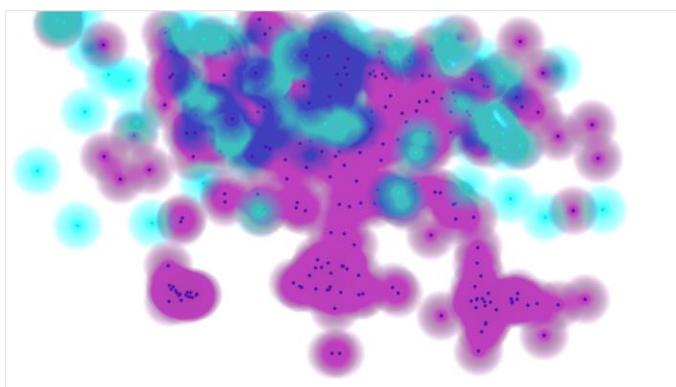


Julia

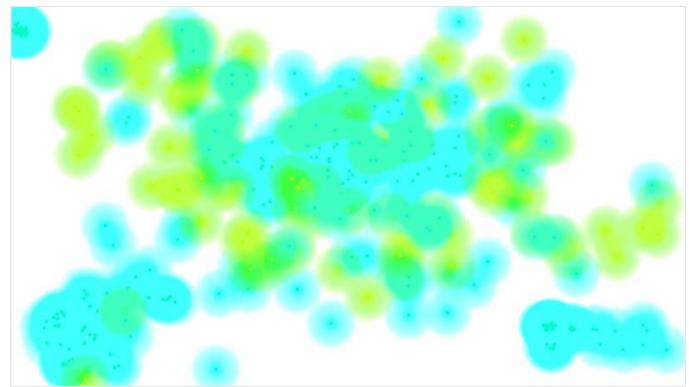
Andi



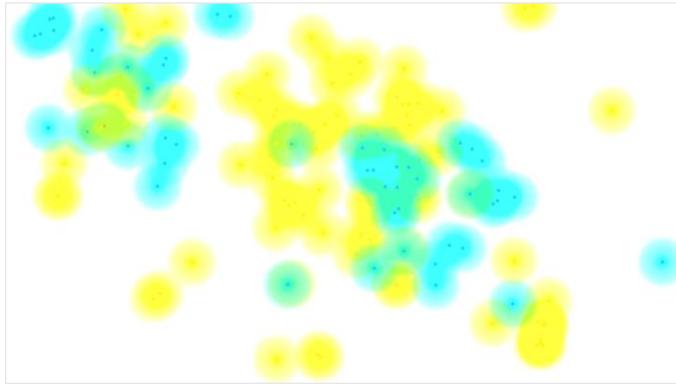
Fiona



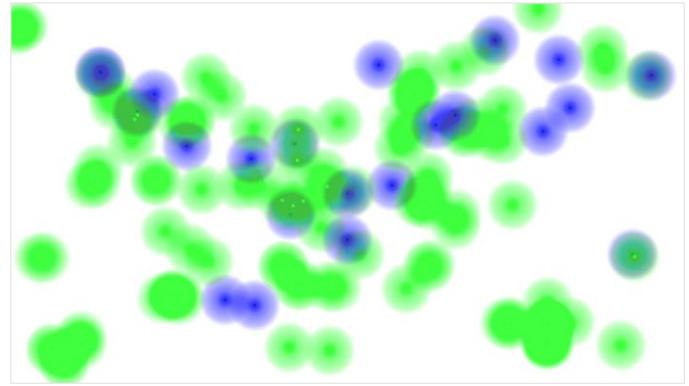
Kyo



Gabriela



Simon



Julia

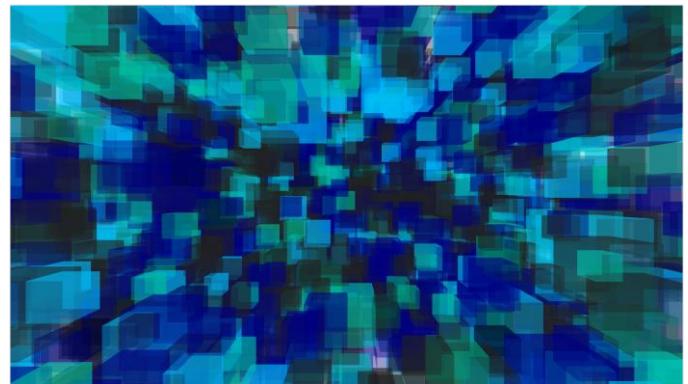
Andi

9.3.3 Programm 3

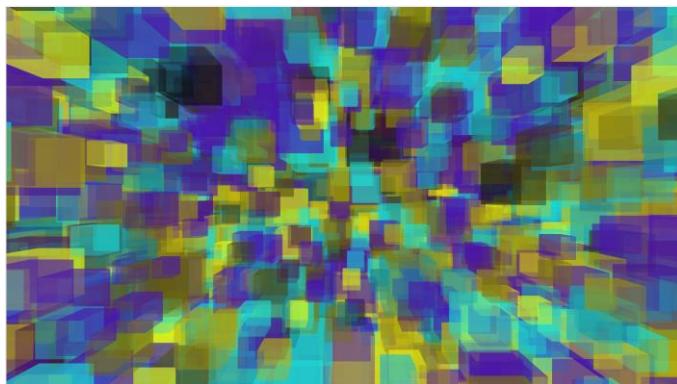
Aufsatz



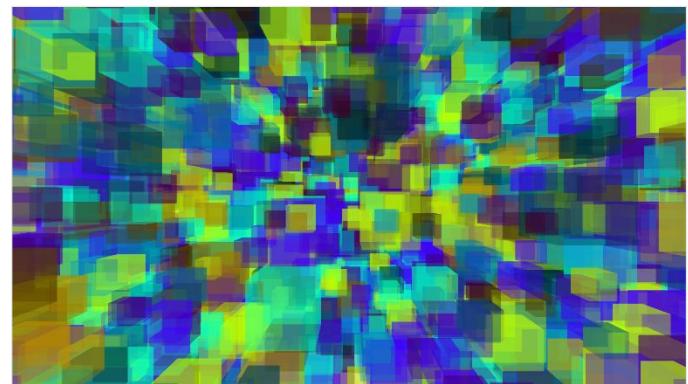
Fiona



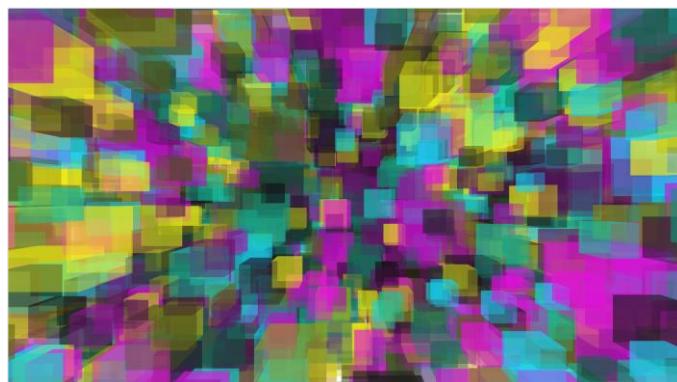
Kyo



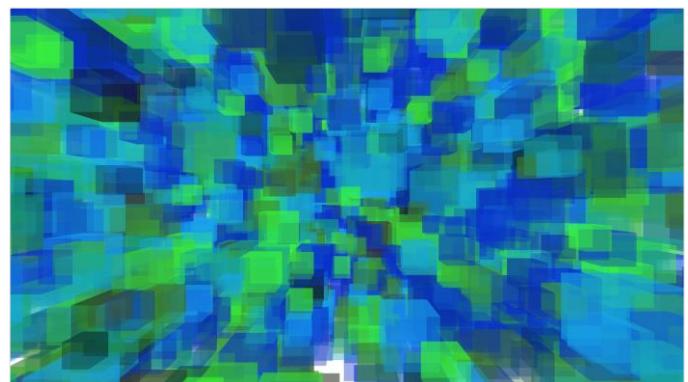
Gabriela



Simon

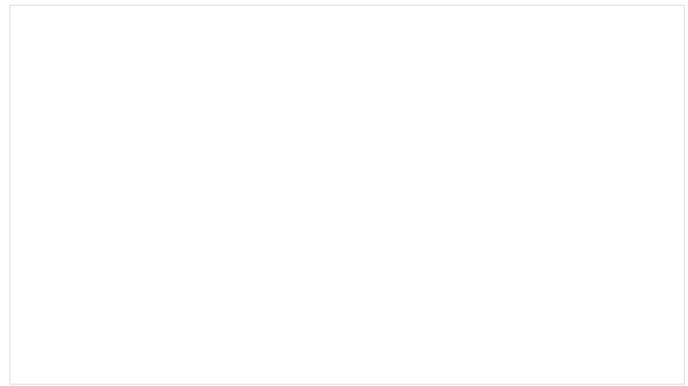
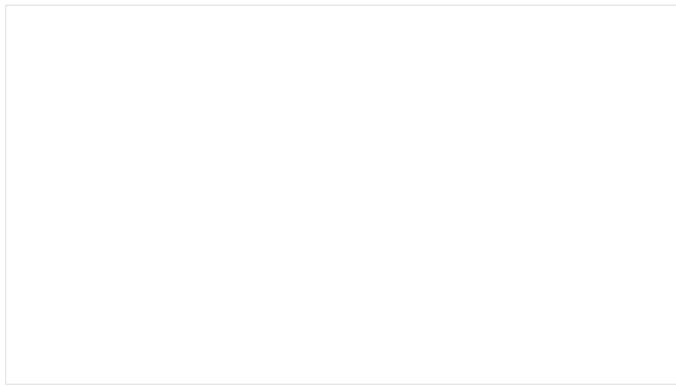


Julia

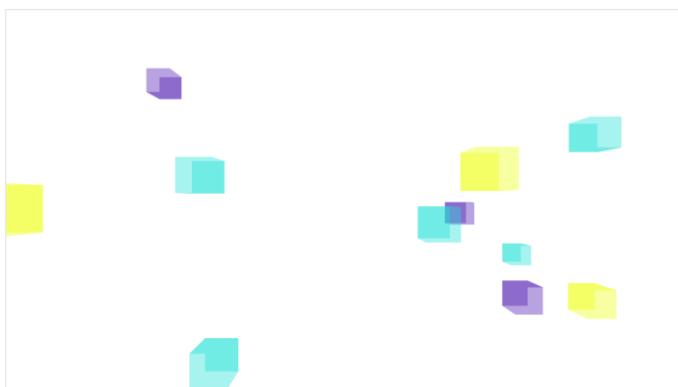


Andi

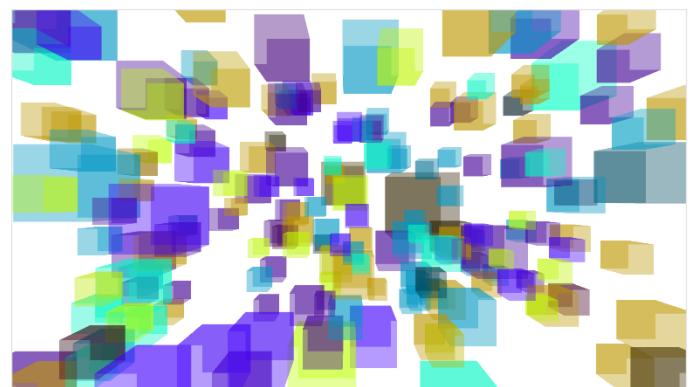
Bild



Fiona



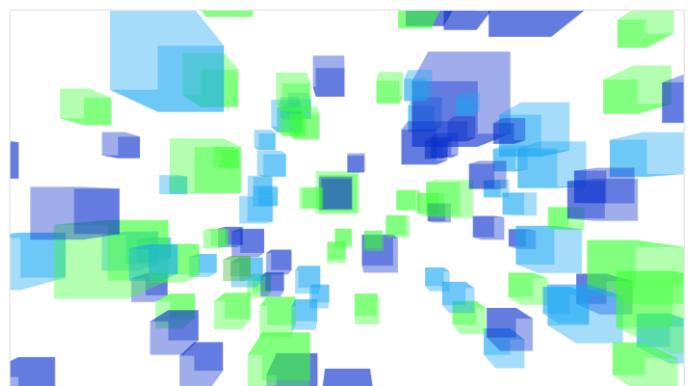
Kyo



Gabriela



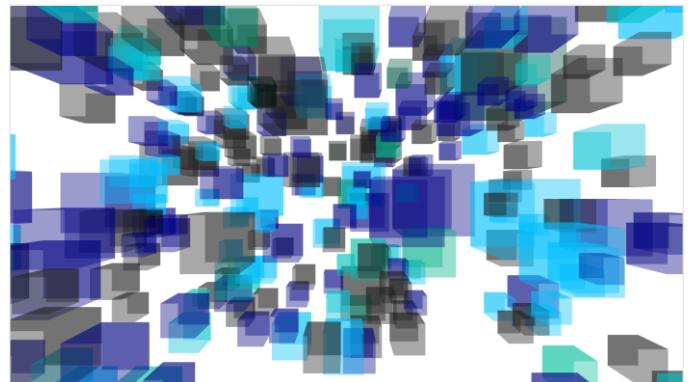
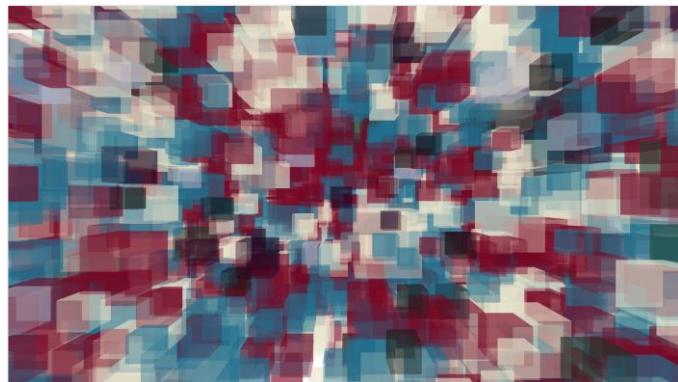
Simon



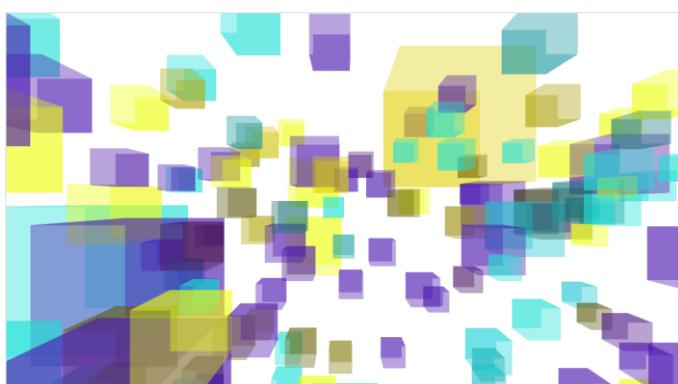
Julia

Andi

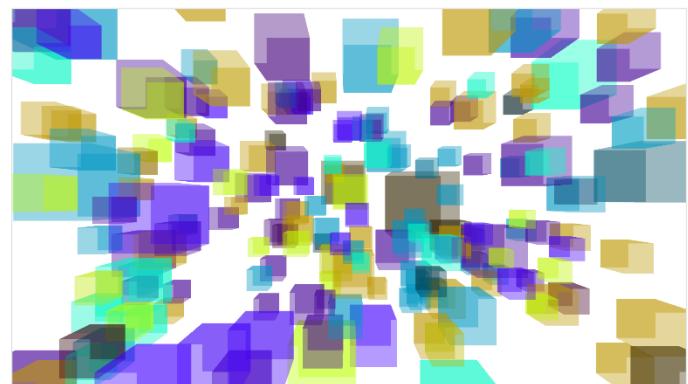
Internet



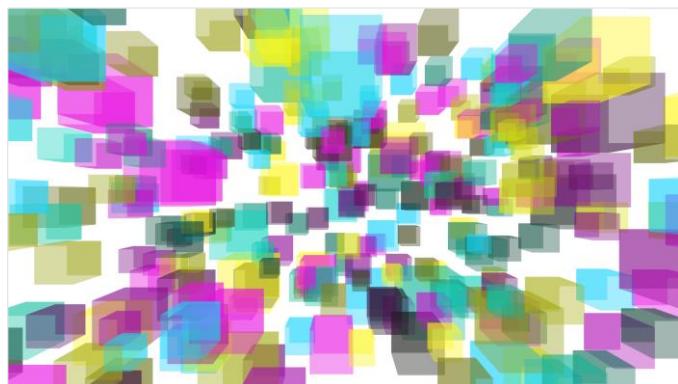
Fiona



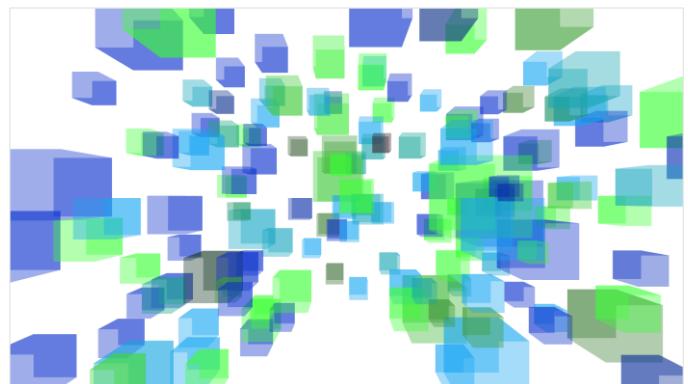
Kyo



Gabriela



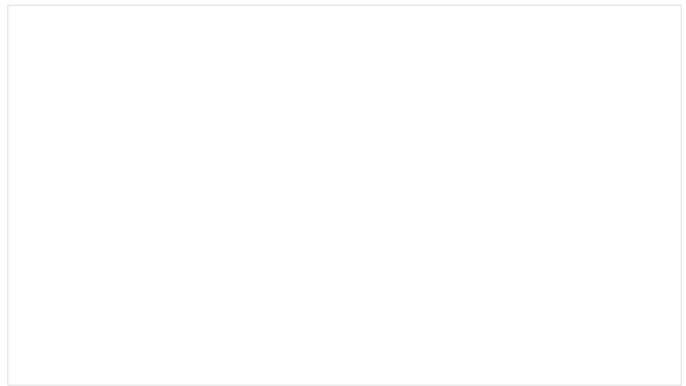
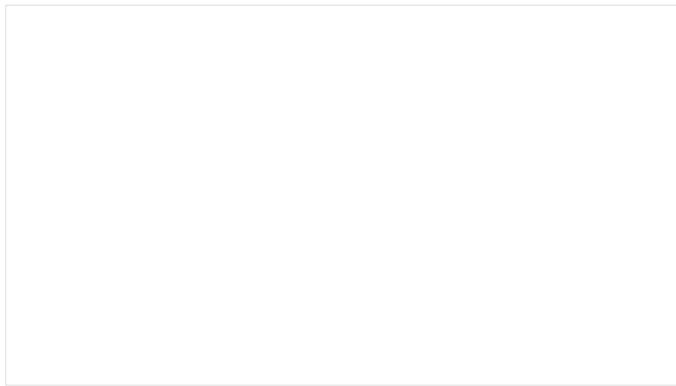
Simon



Julia

Andi

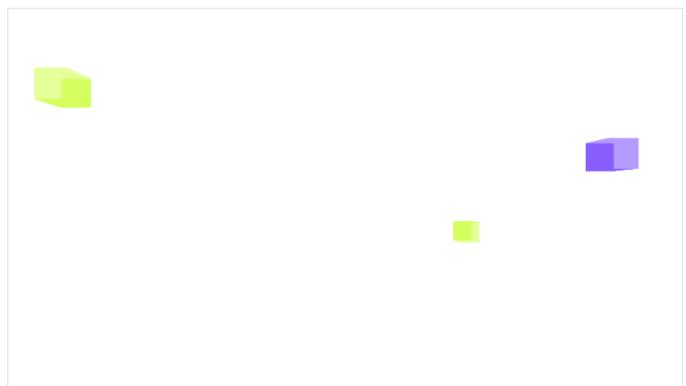
Spiel



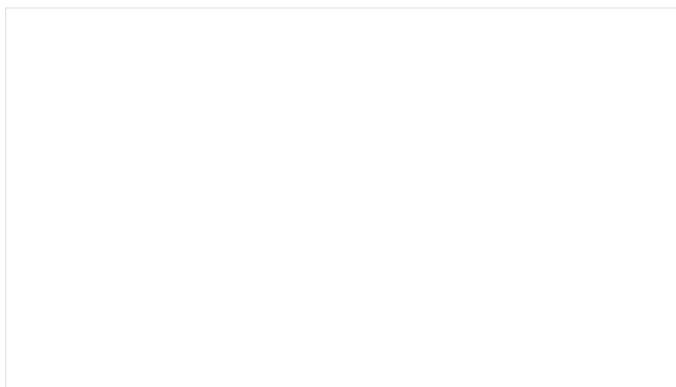
Fiona



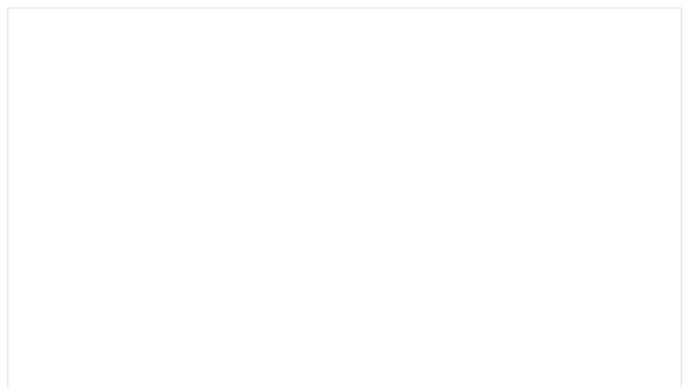
Kyo



Gabriela



Simon



Julia

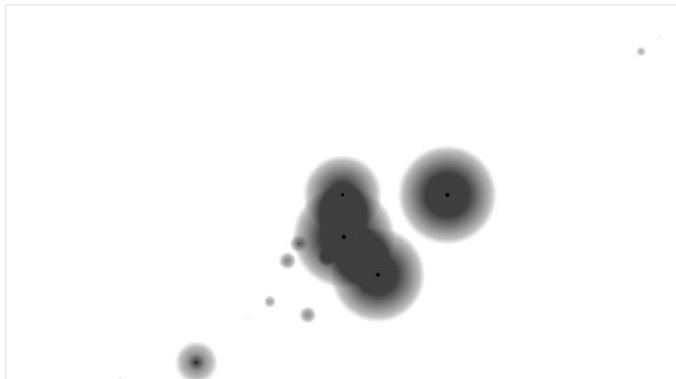


Andi



9.3.4 Programm 4

Aufsatz



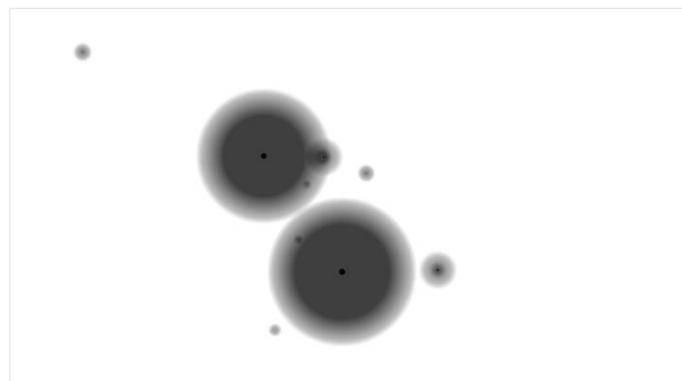
Fiona



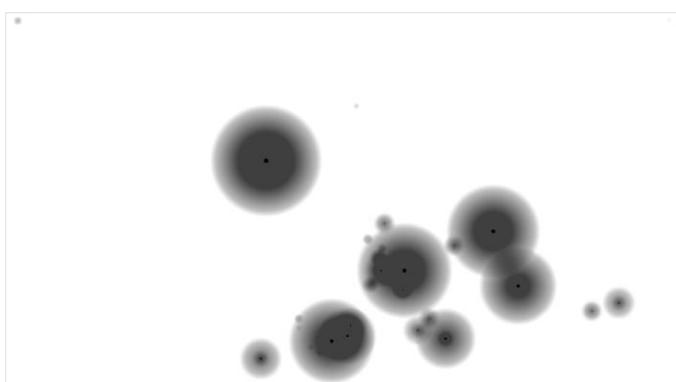
Kyo



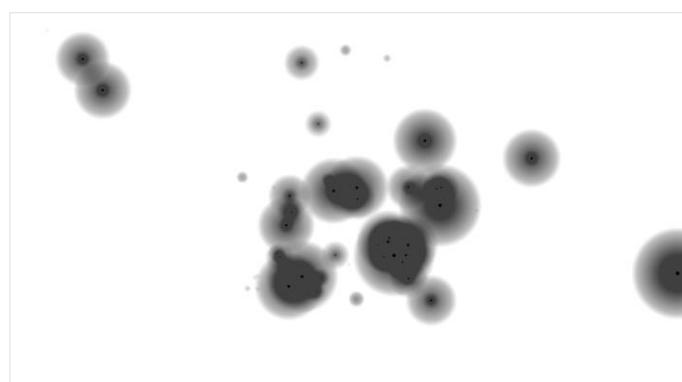
Gabriela



Simon



Julia



Andi

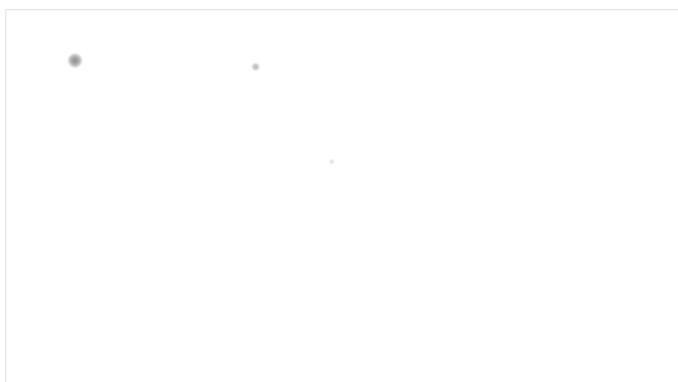
Bild



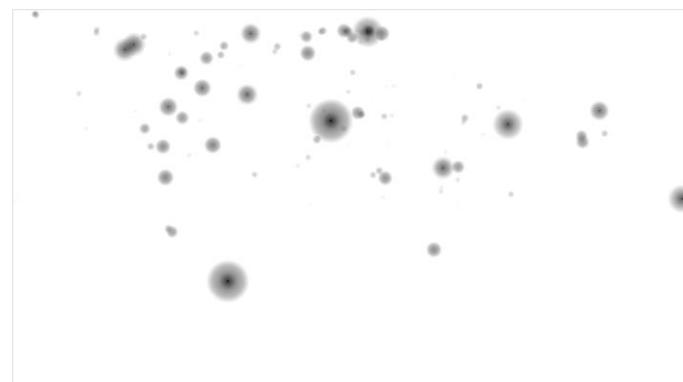
Fiona



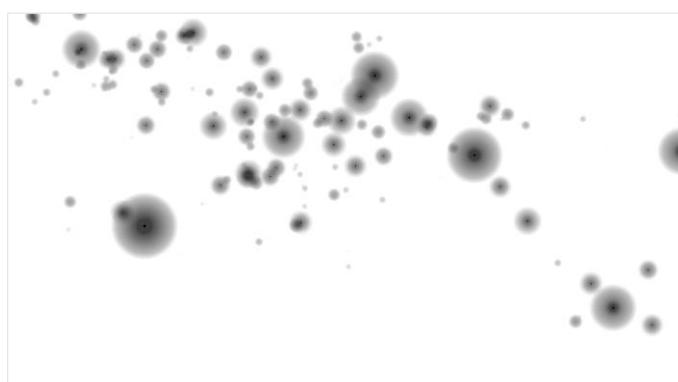
Kyo



Gabriela



Simon

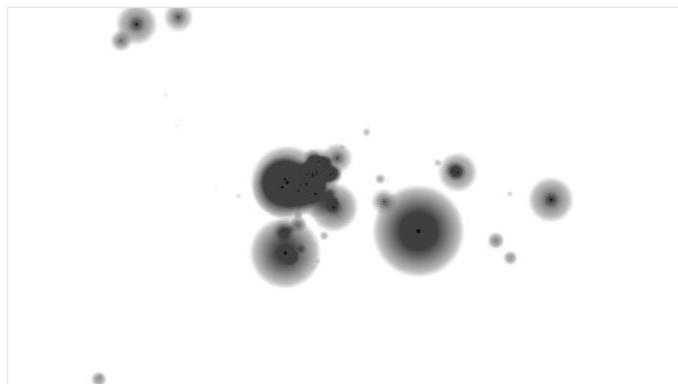


Julia

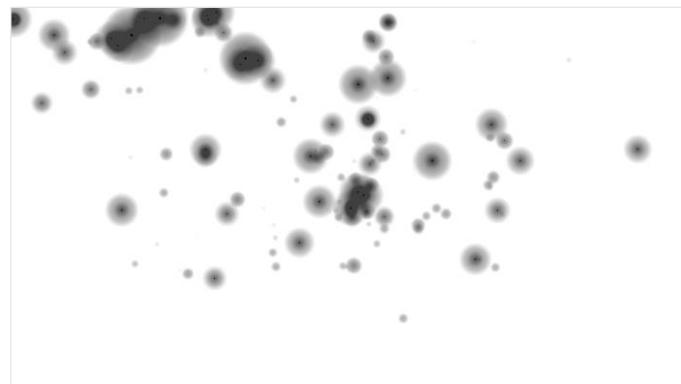


Andi

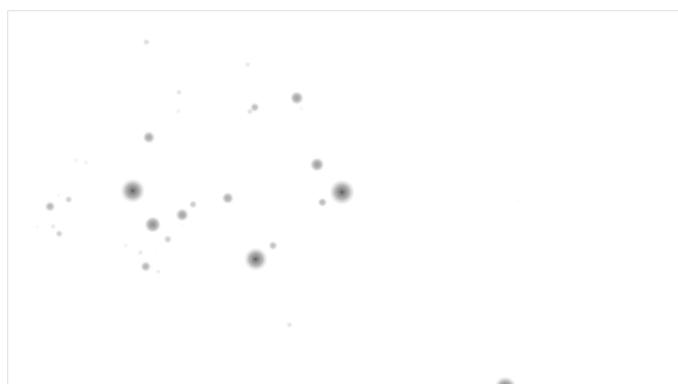
Internet



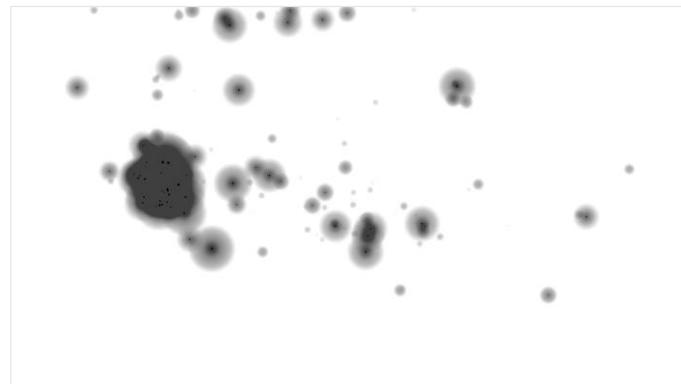
Fiona



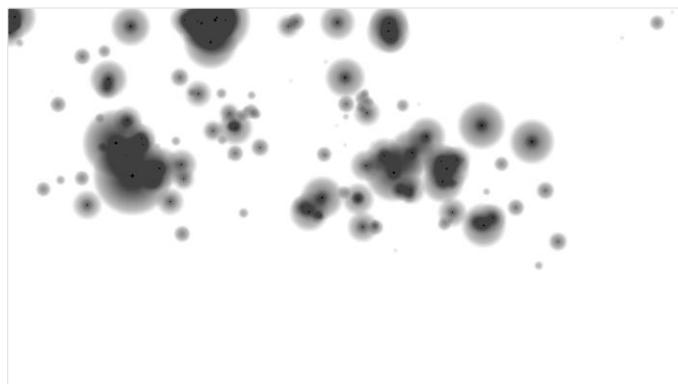
Kyo



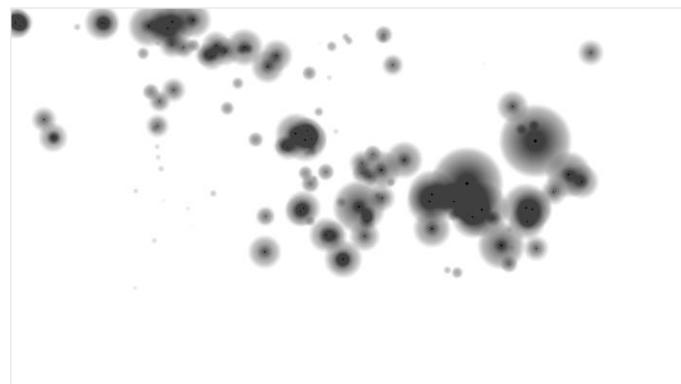
Gabriela



Simon



Julia



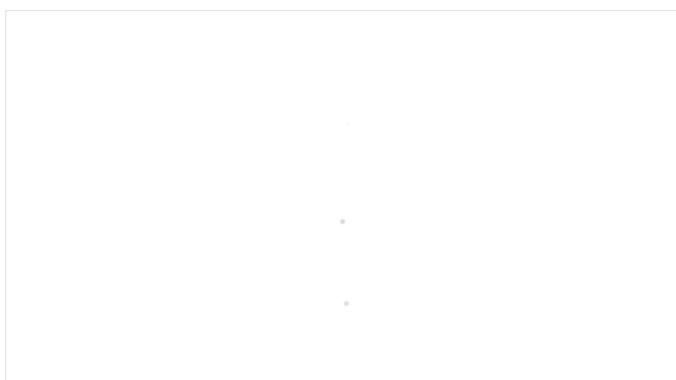
Andi

Spiel



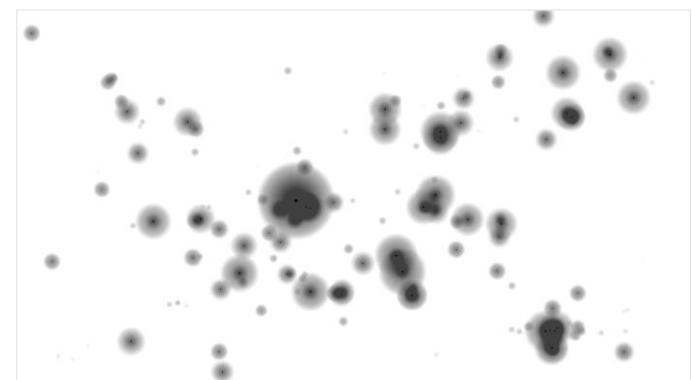
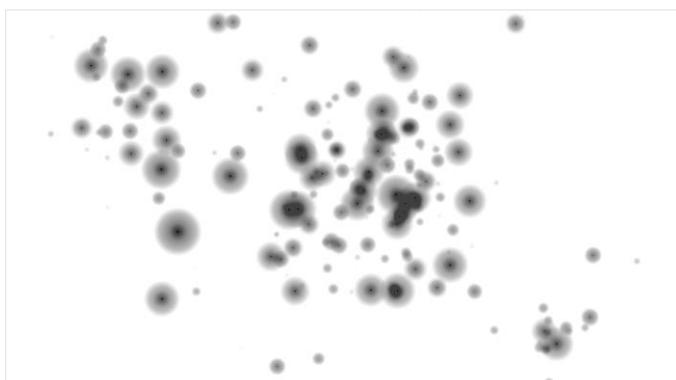
Fiona

Kyo



Gabriela

Simon

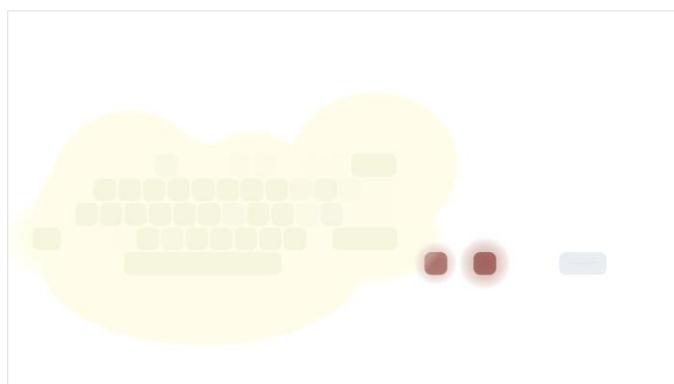


Julia

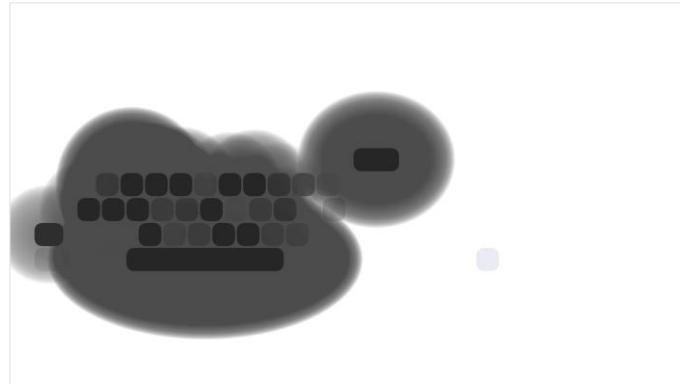
Andi

9.3.5 Programm 5

Aufsatz



Fiona



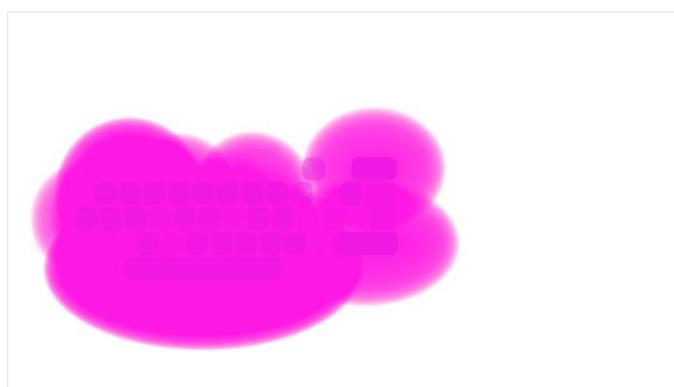
Kyo



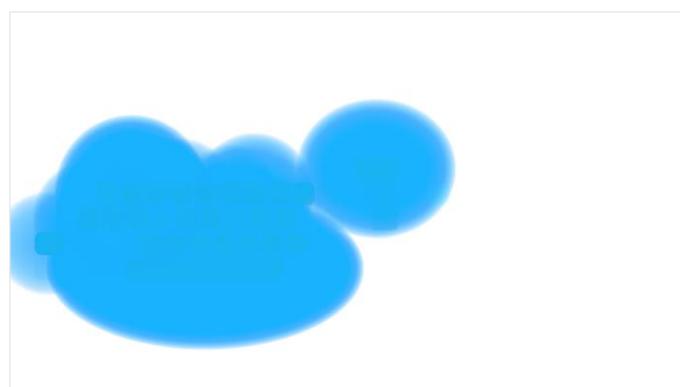
Gabriela



Simon

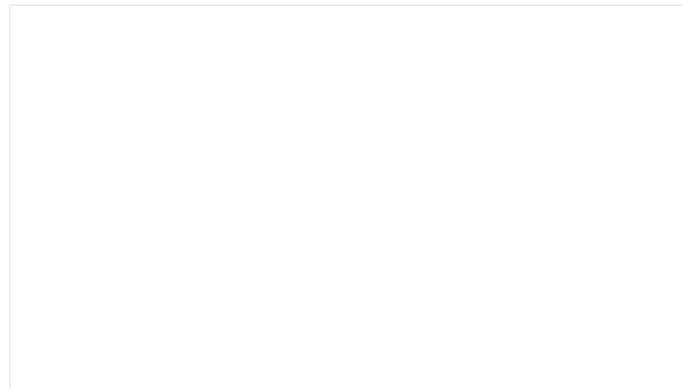
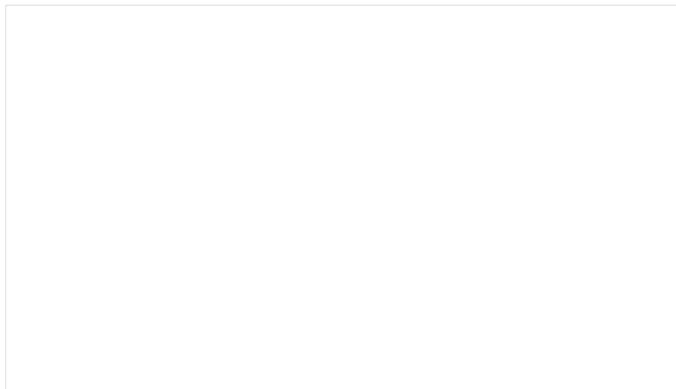


Julia



Andi

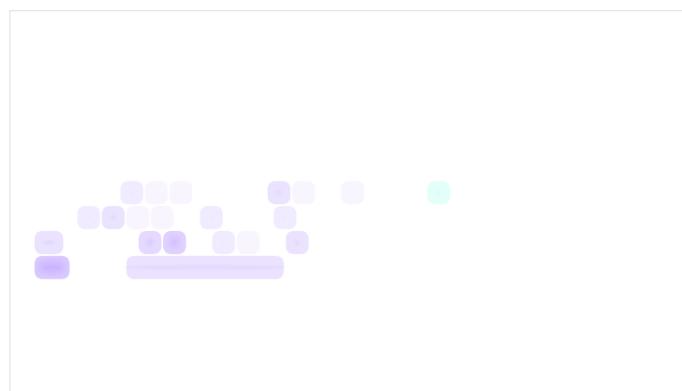
Bild



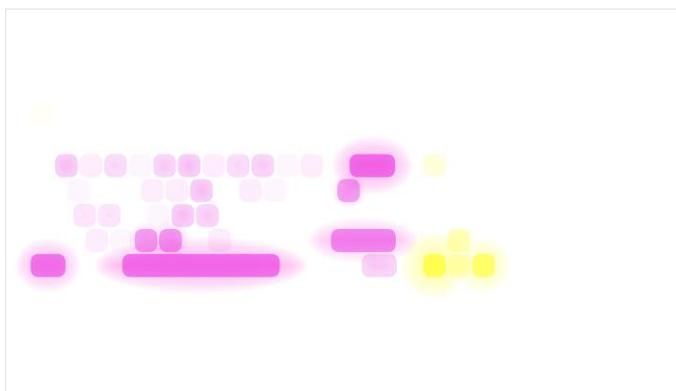
Fiona



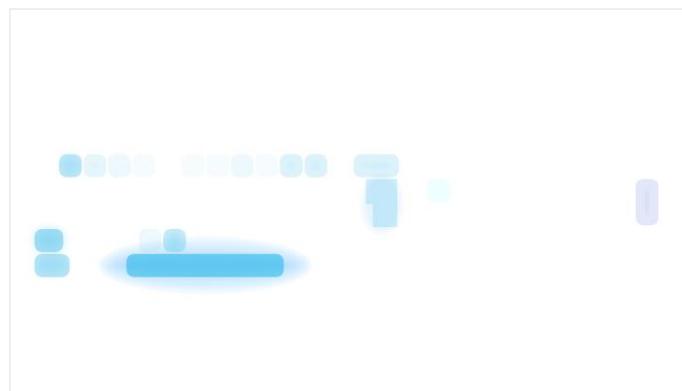
Kyo



Gabriela



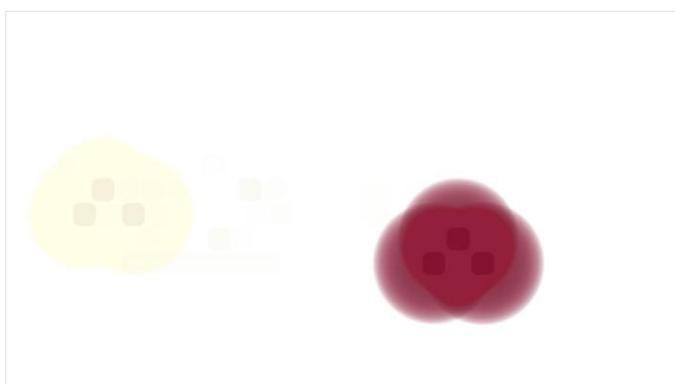
Simon



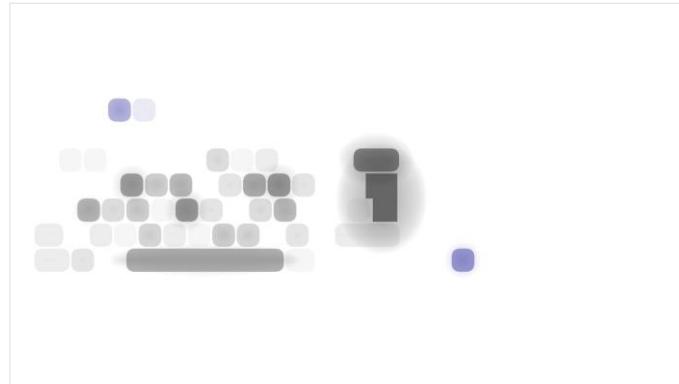
Julia

Andi

Internet



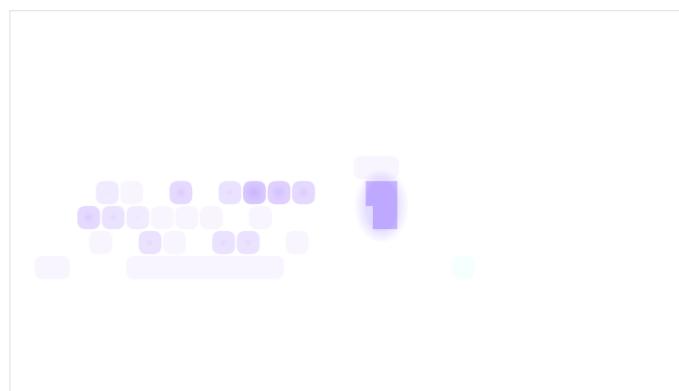
Fiona



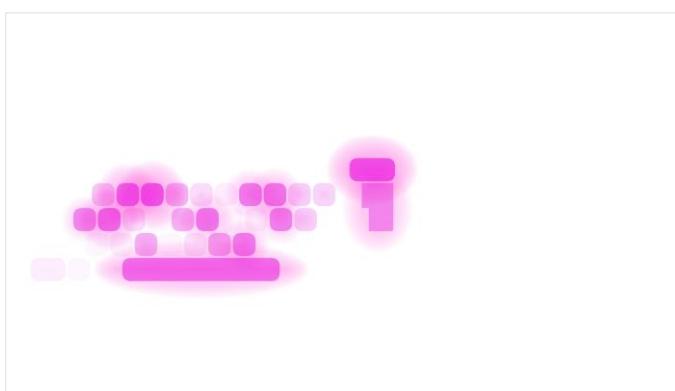
Kyo



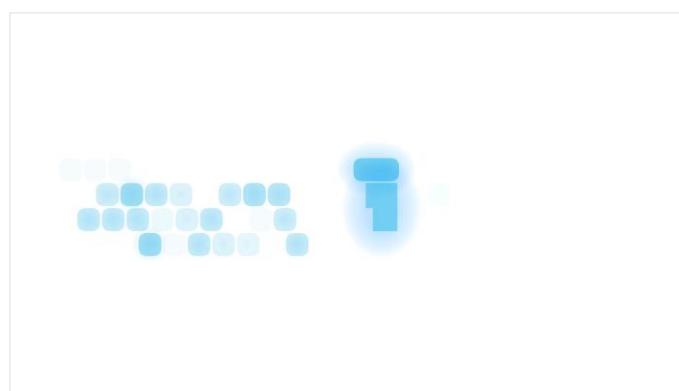
Gabriela



Simon

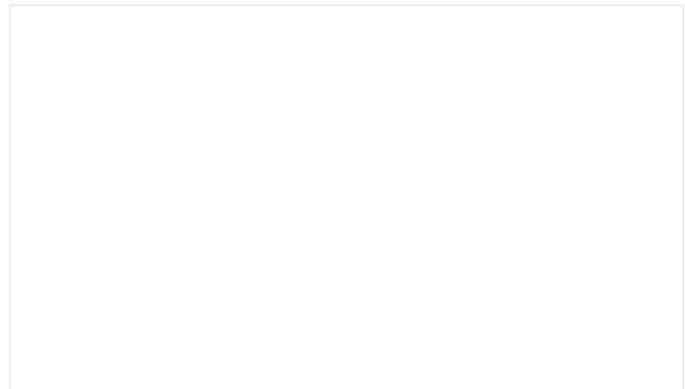
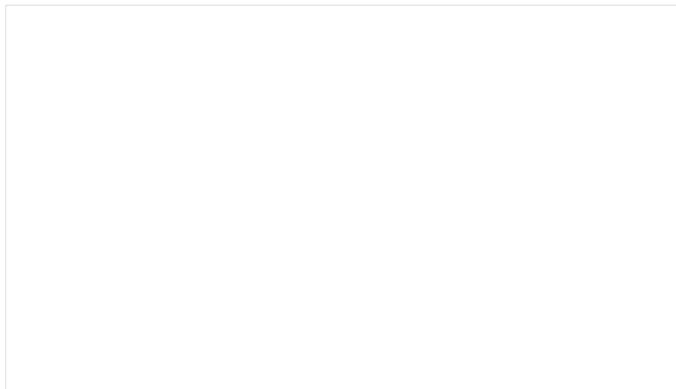


Julia



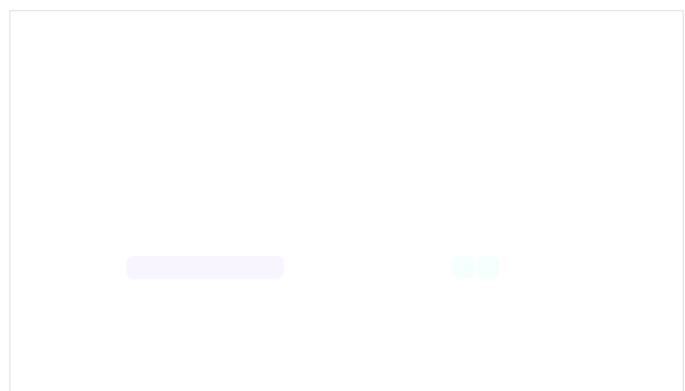
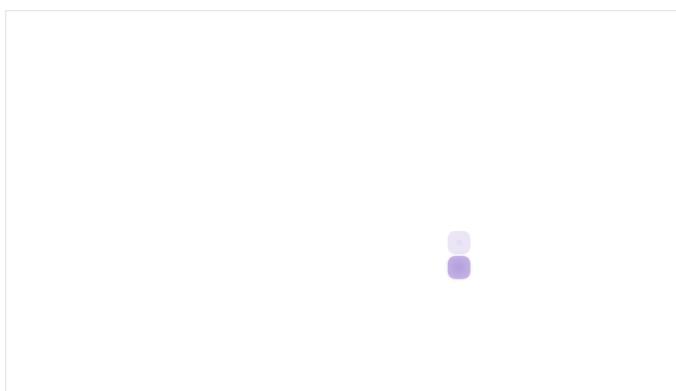
Andi

Spiel



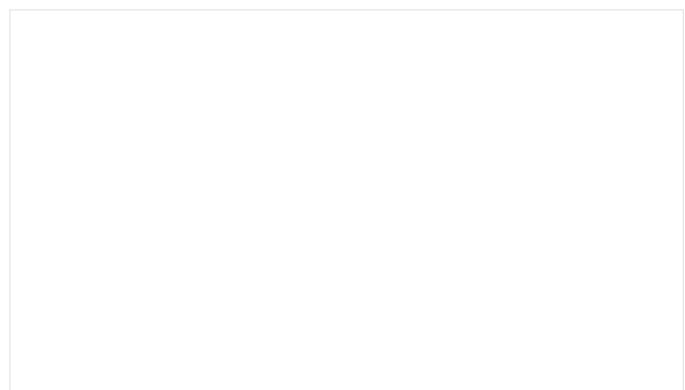
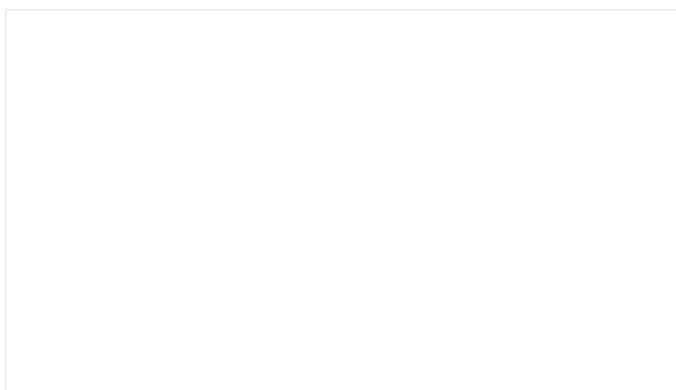
Fiona

Kyo



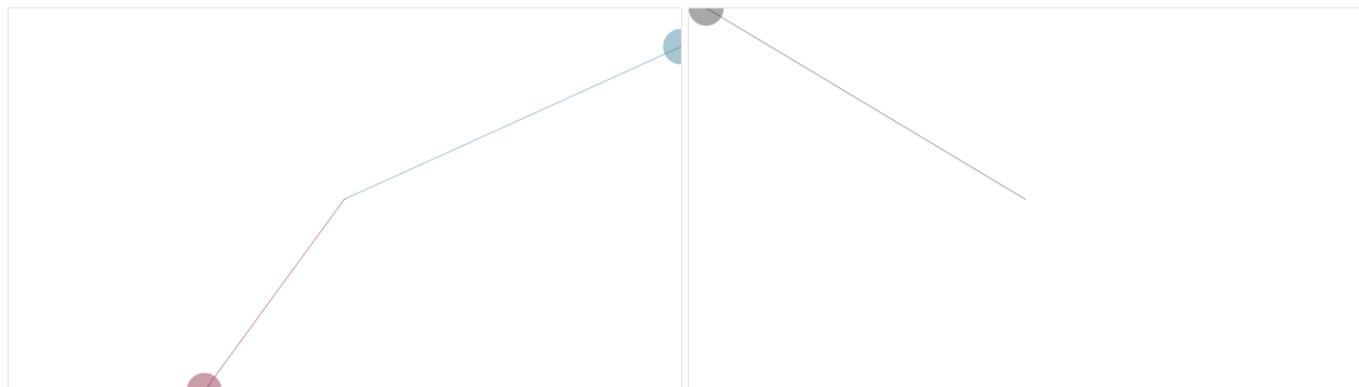
Gabriela

Simon



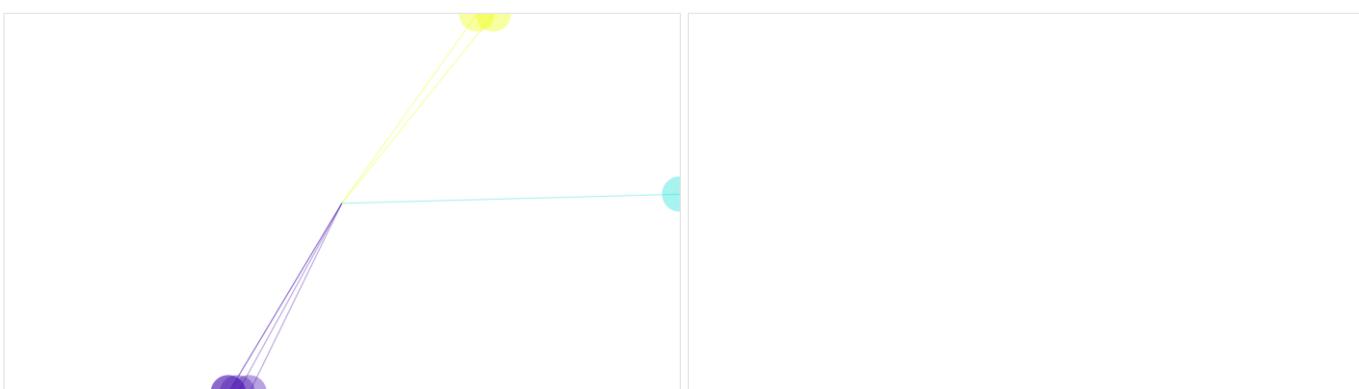
Julia

Andi

9.3.6 Programm 6**Aufsatz**

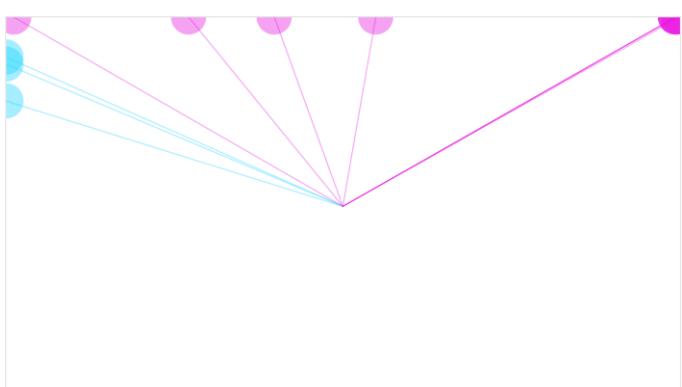
Fiona

Kyo



Gabriela

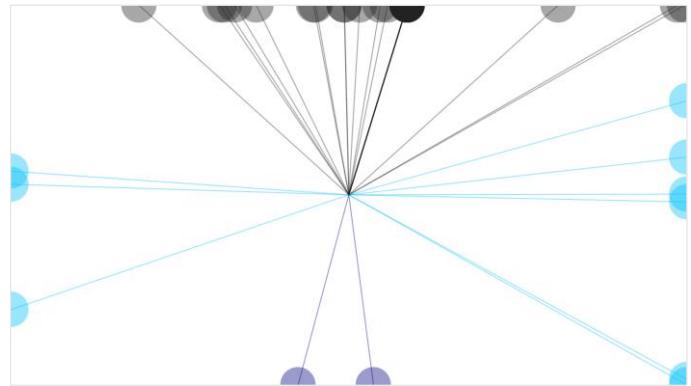
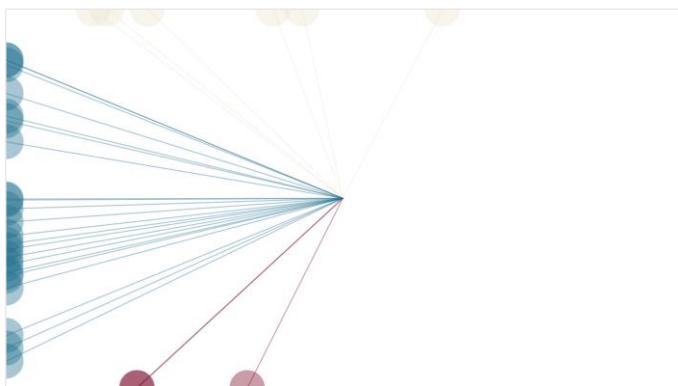
Simon



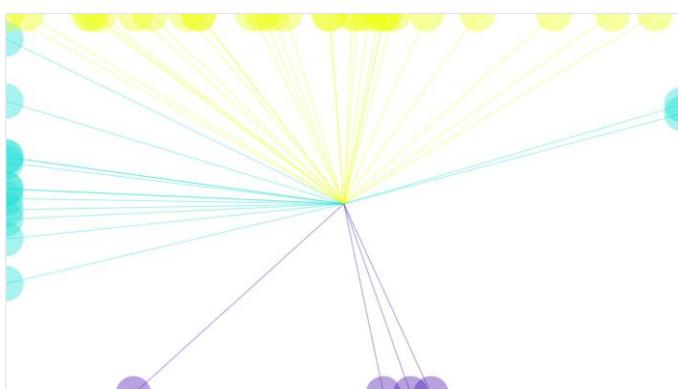
Julia

Andi

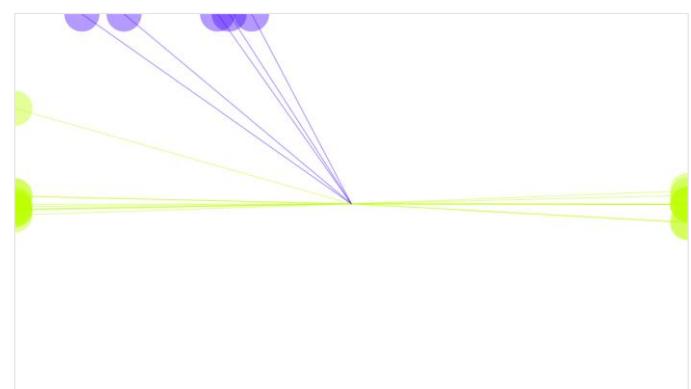


Bild

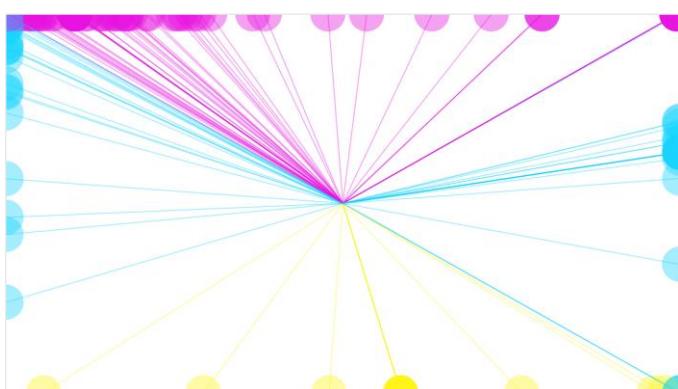
Fiona



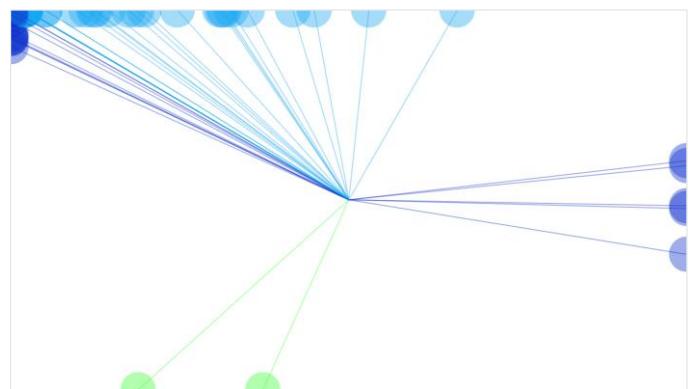
Kyo



Gabriela

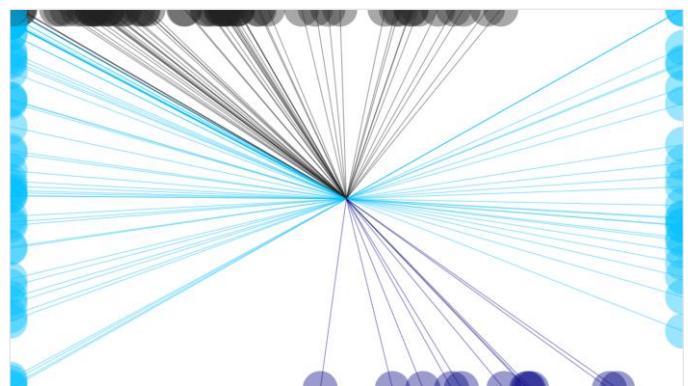
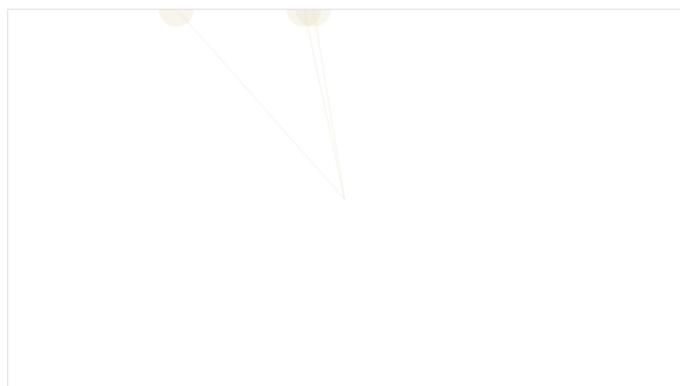


Simon

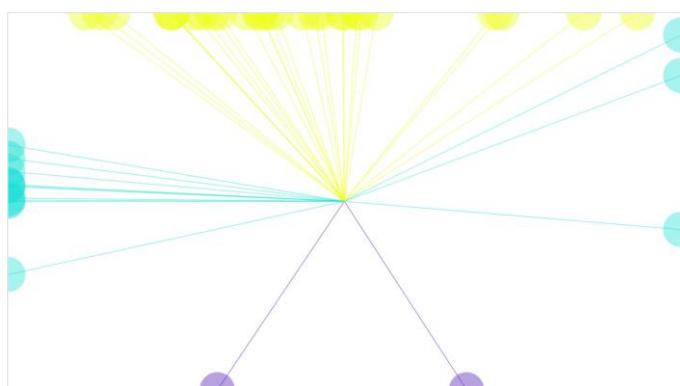


Julia

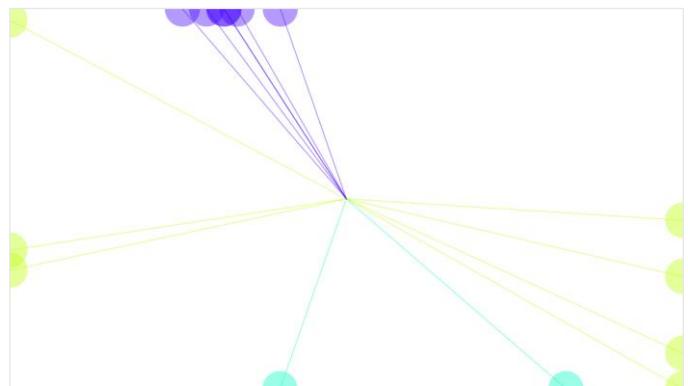
Andi

Internet

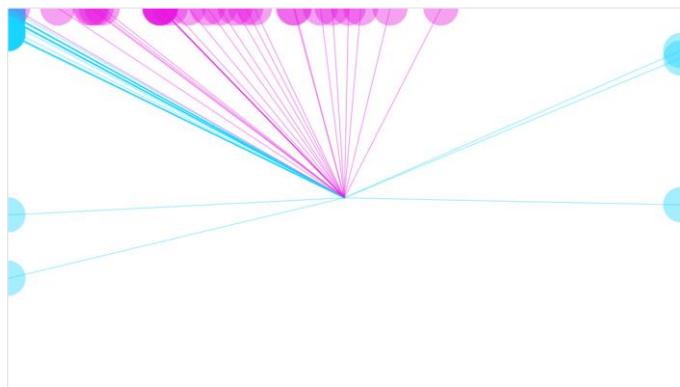
Fiona



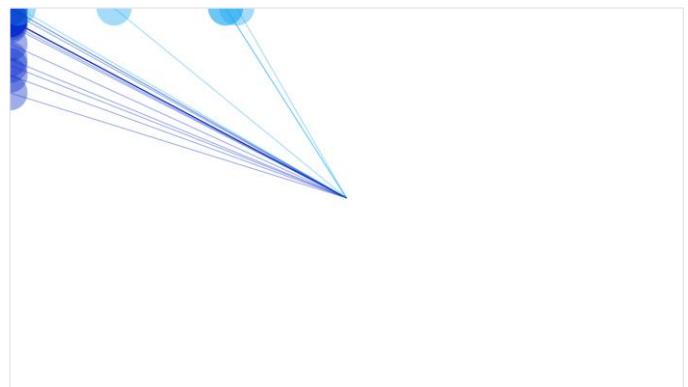
Kyo



Gabriela



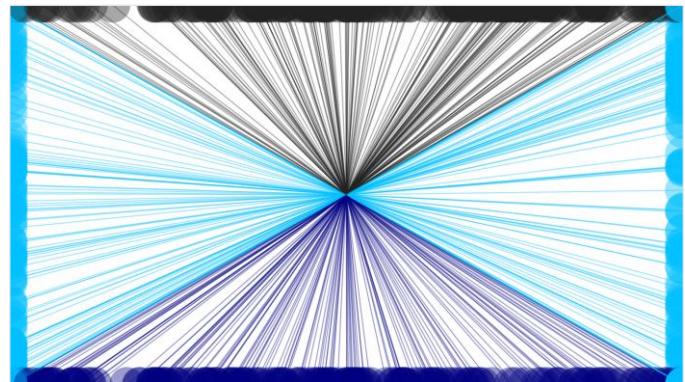
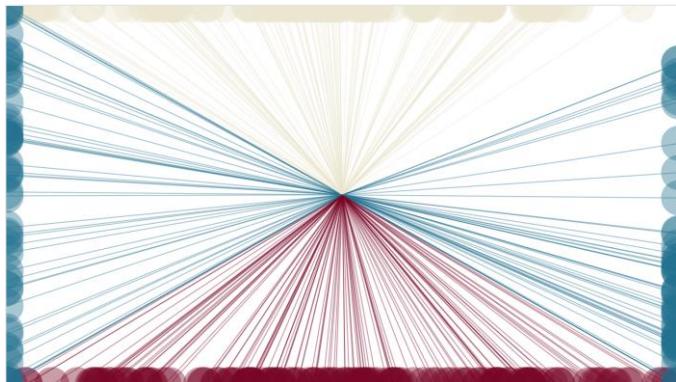
Simon



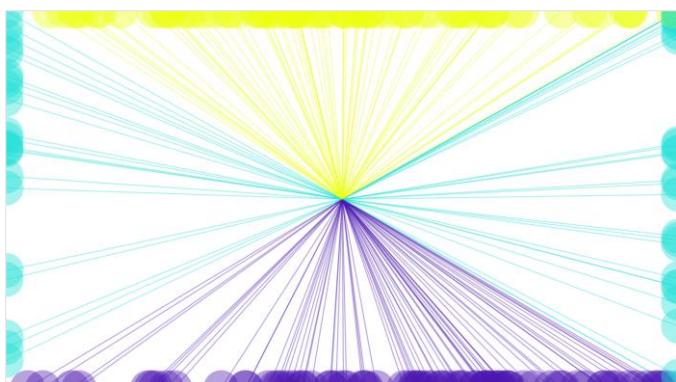
Julia

Andi

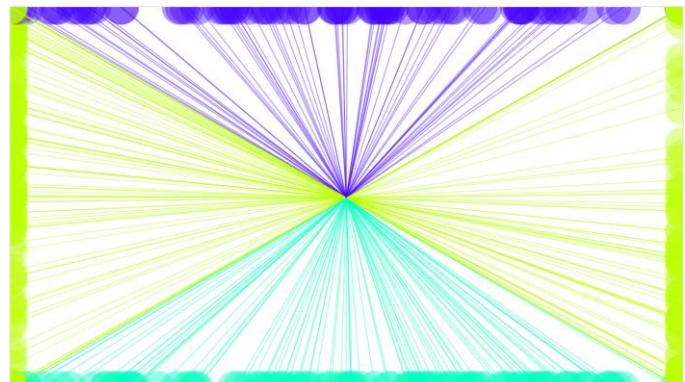
Spiel



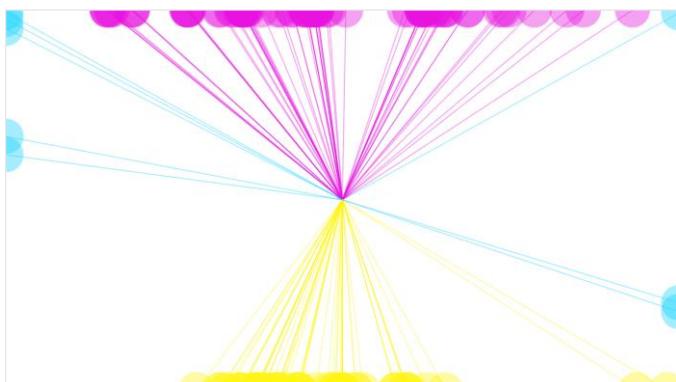
Fiona



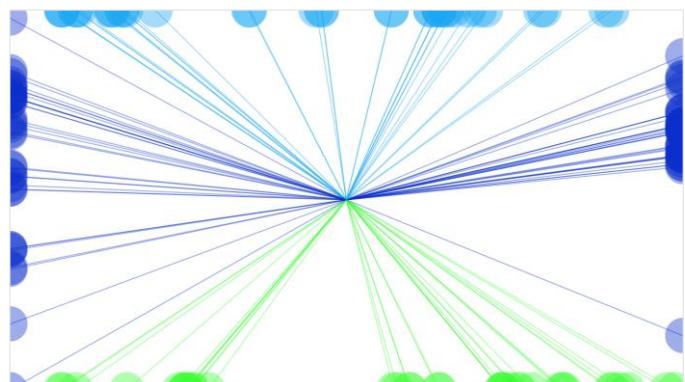
Kyo



Gabriela



Simon

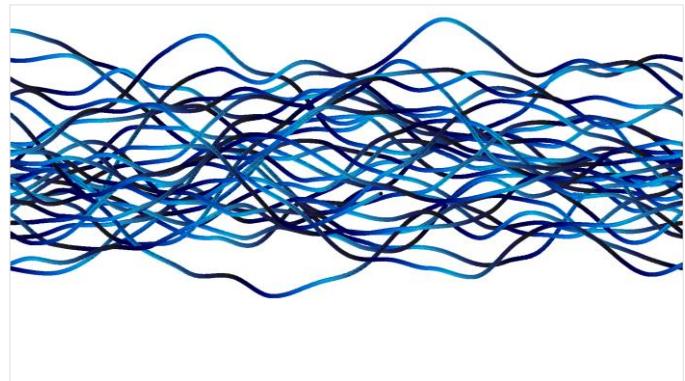
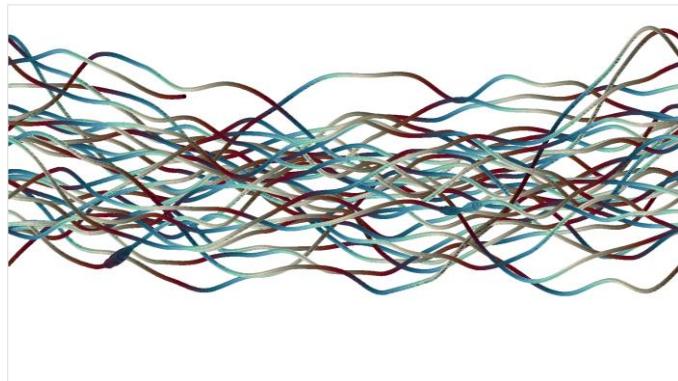


Julia

Andi

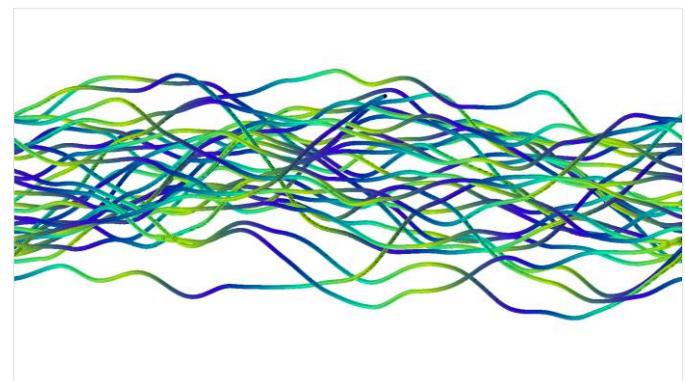
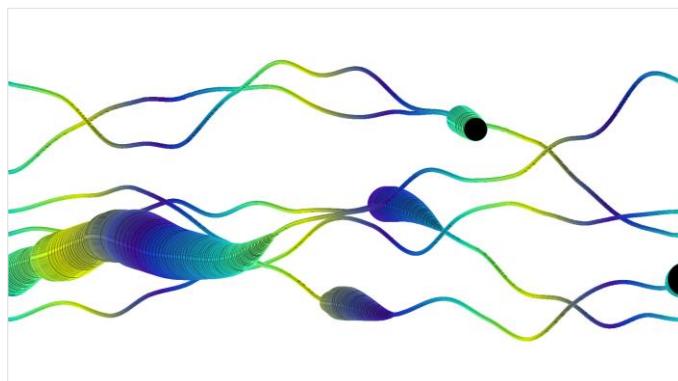
9.3.7 Programm 7

Aufsatz



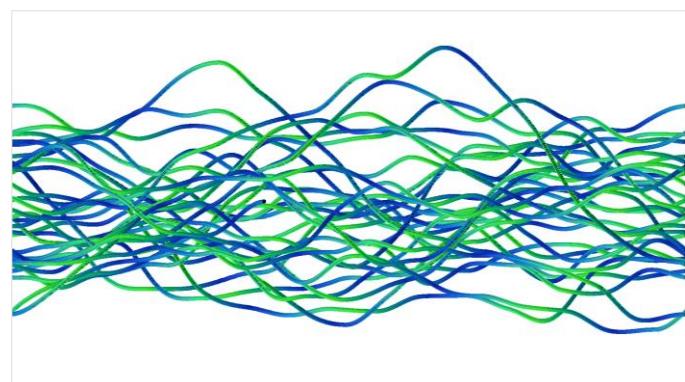
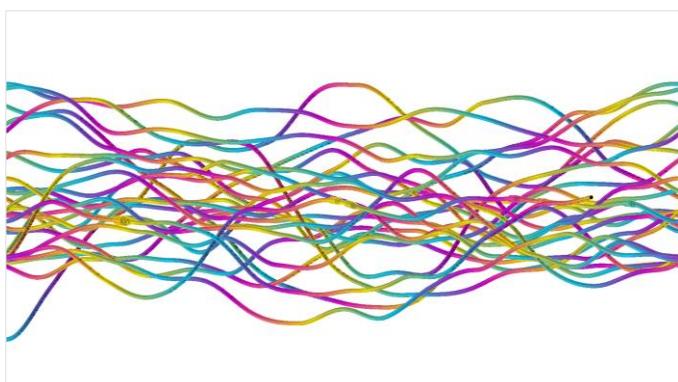
Fiona

Kyo



Gabriela

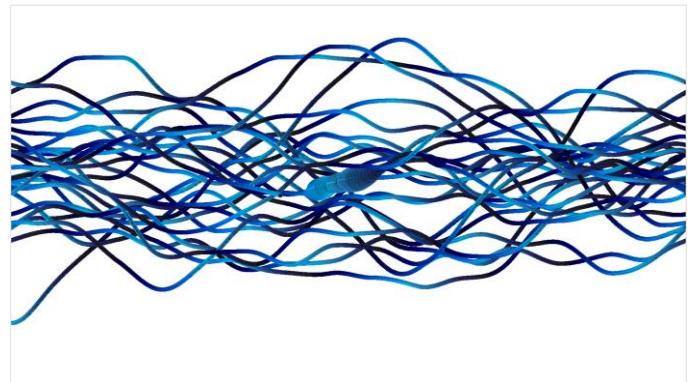
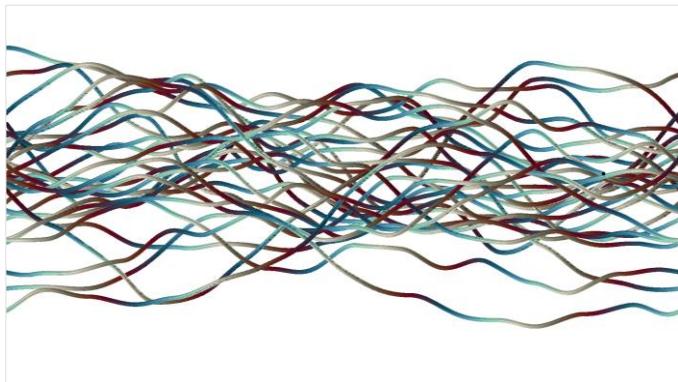
Simon



Julia

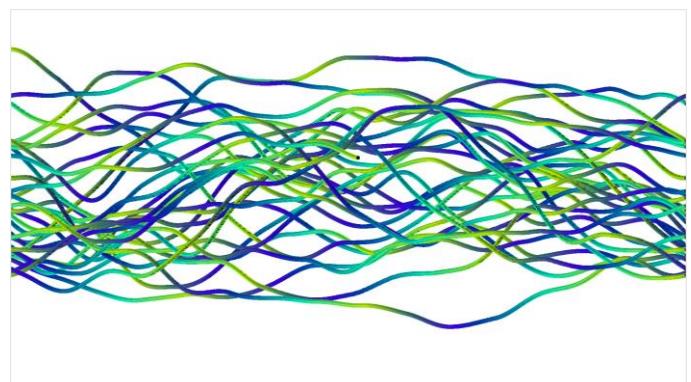
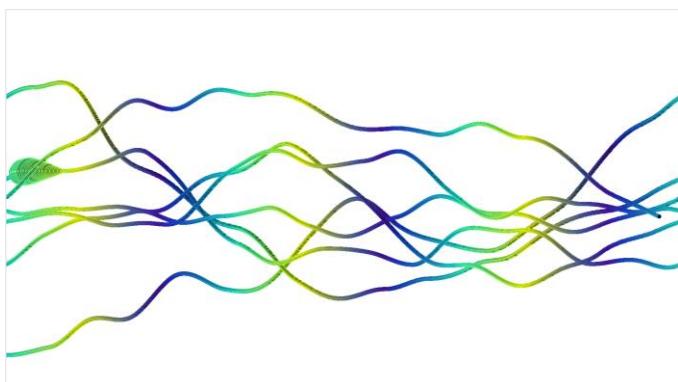
Andi

Bild



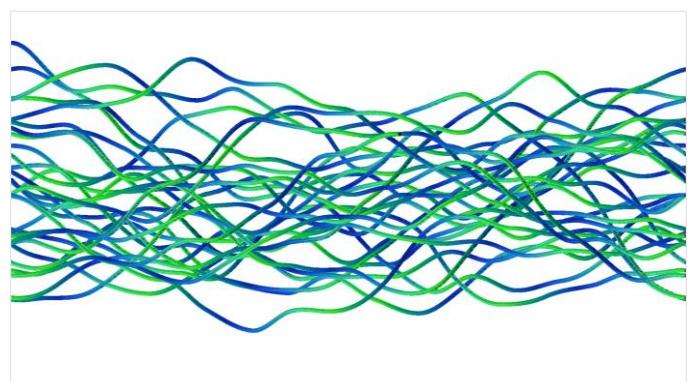
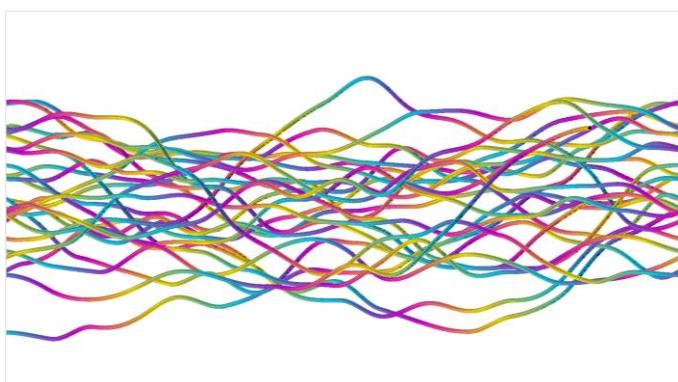
Fiona

Kyo



Gabriela

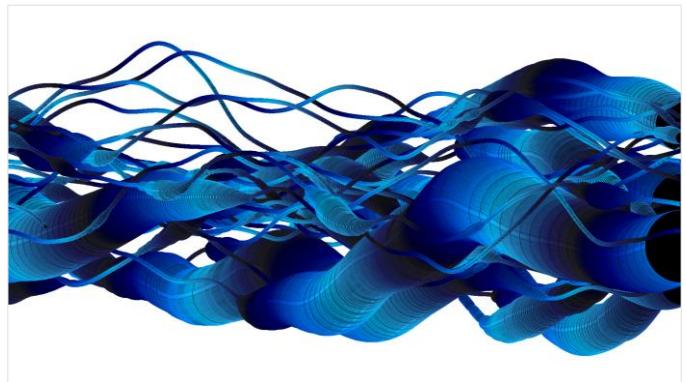
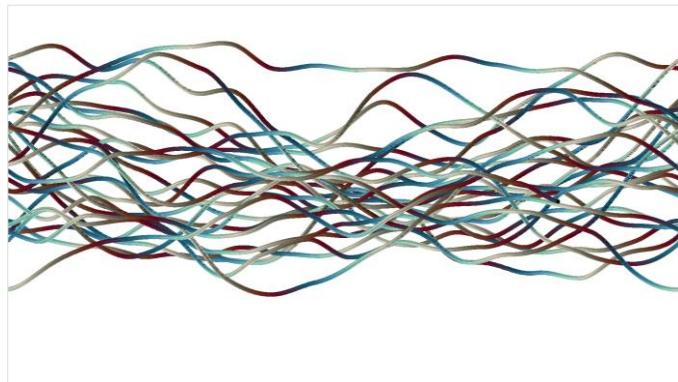
Simon



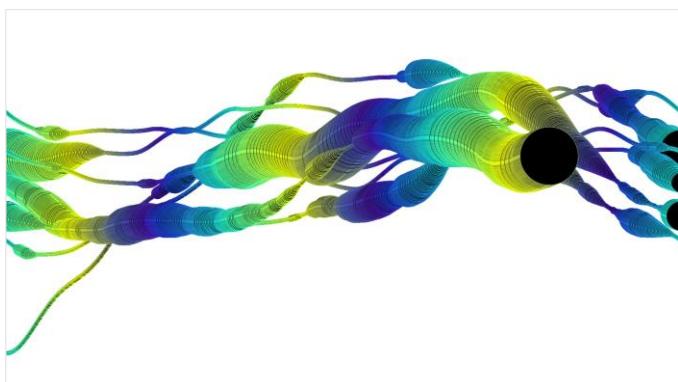
Julia

Andi

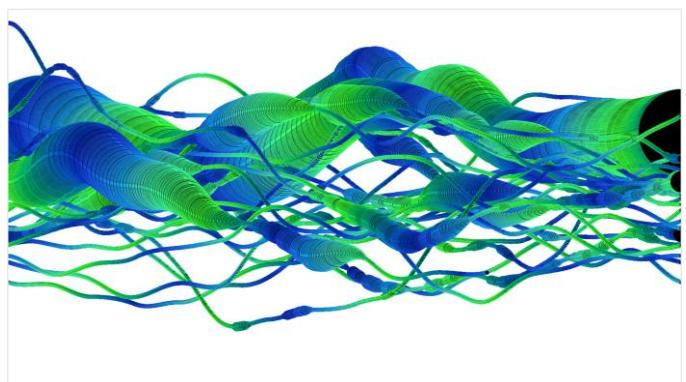
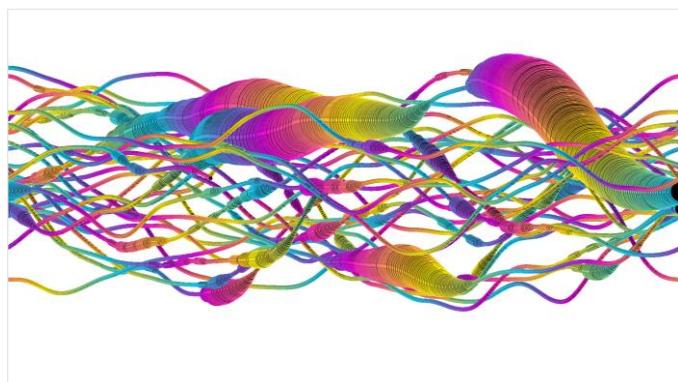
Internet



Fiona



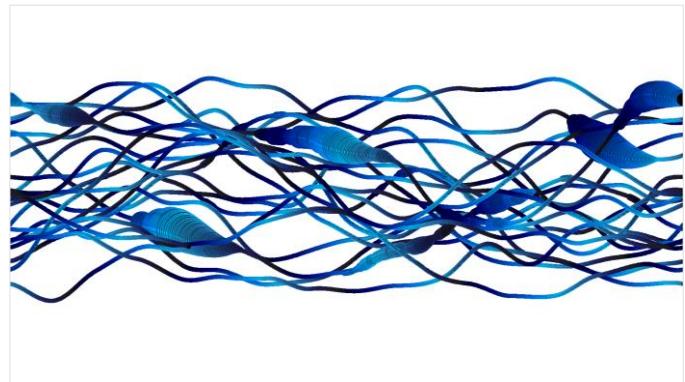
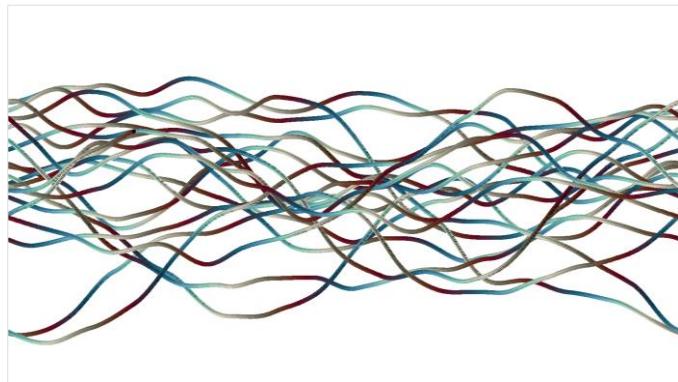
Gabriela



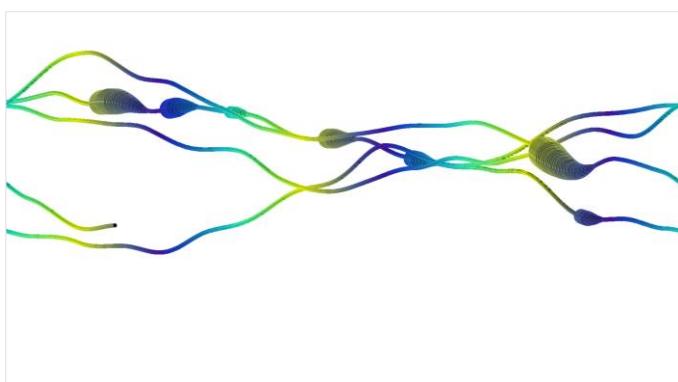
Julia

Simon

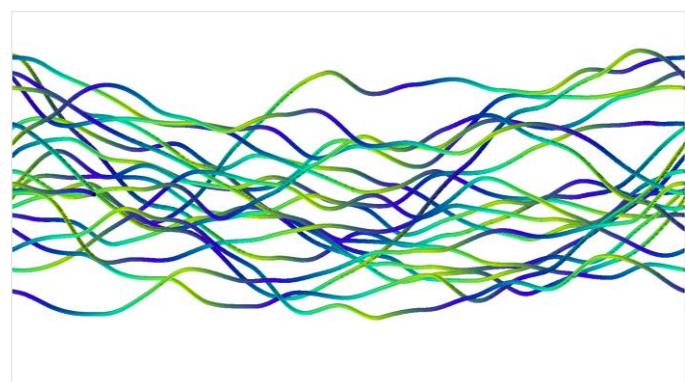
Andi



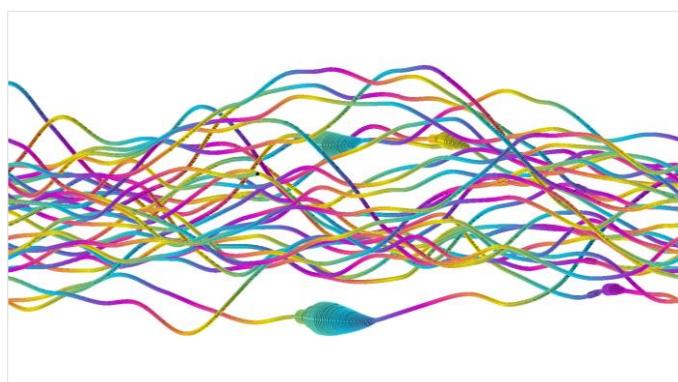
Fiona



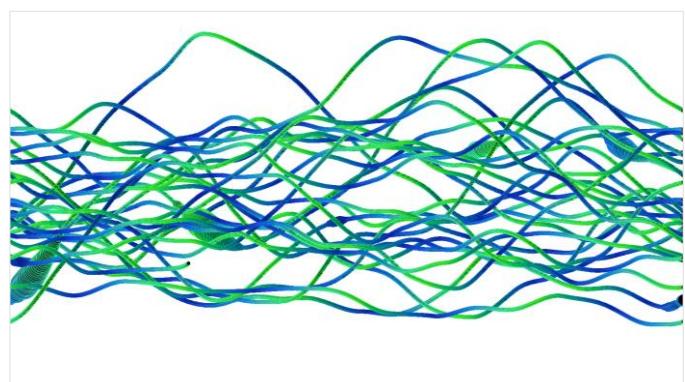
Kyo



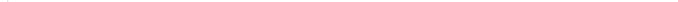
Gabriela



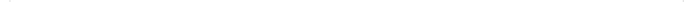
Simon



Julia

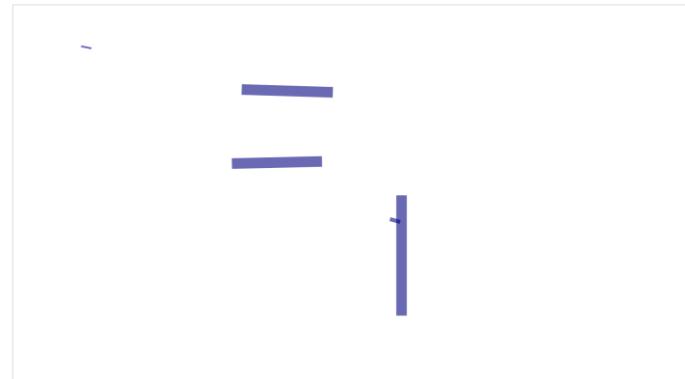
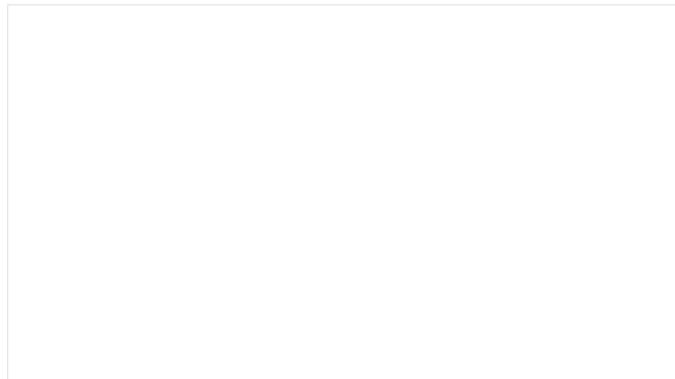


Andi

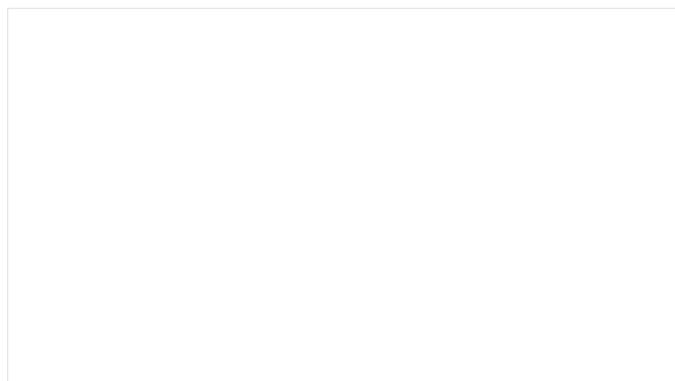


9.3.8 Programm 8

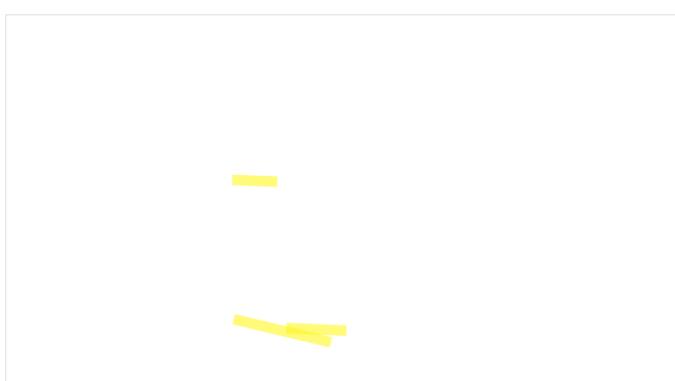
Aufsatz



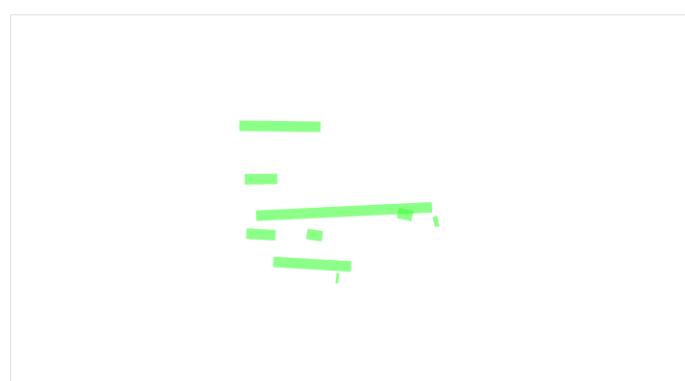
Fiona



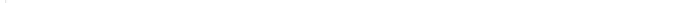
Kyo



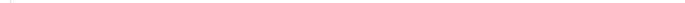
Gabriela



Simon

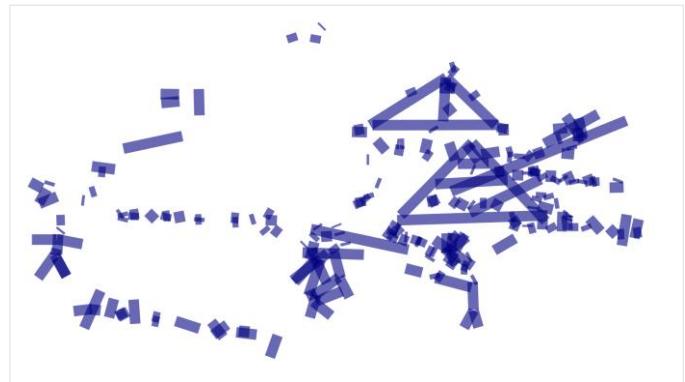
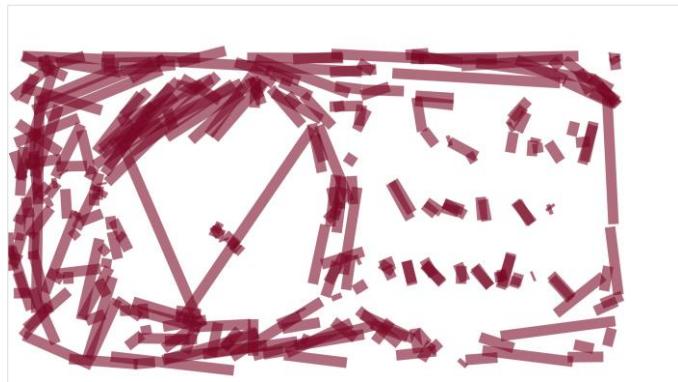


Julia



Andi

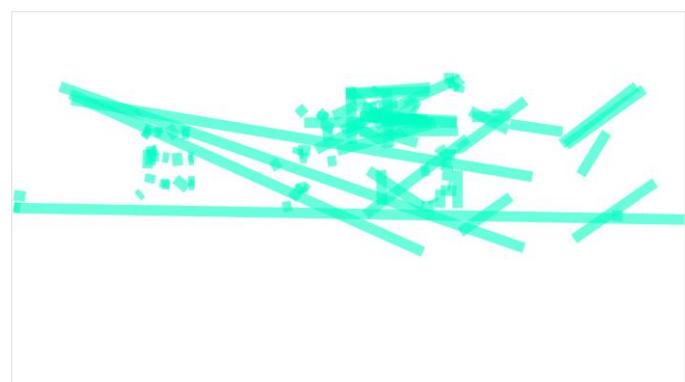
Bild



Fiona



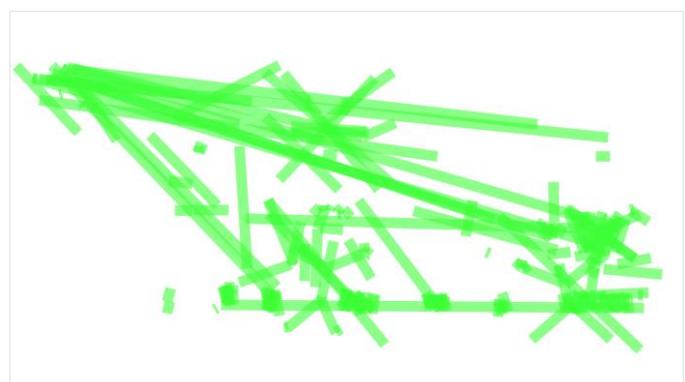
Kyo



Gabriela



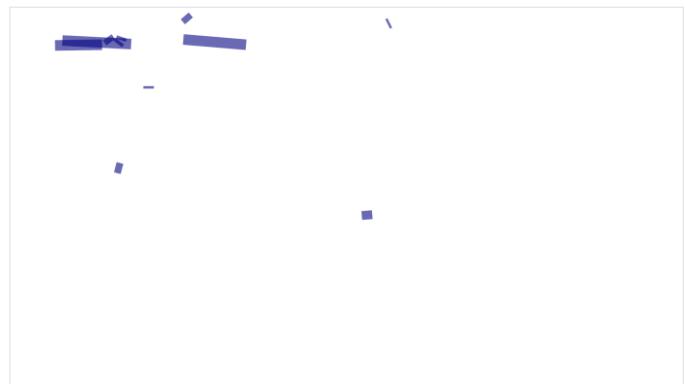
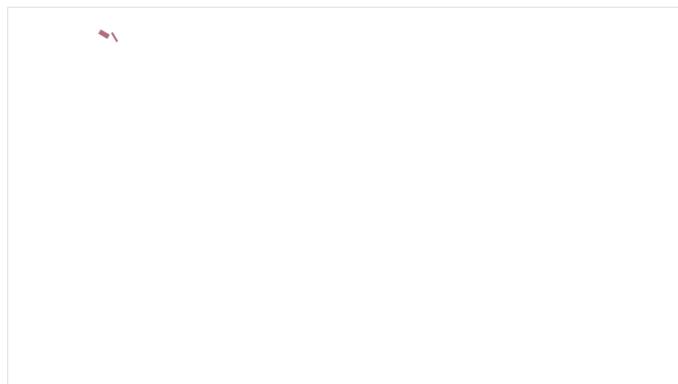
Simon



Julia

Andi

Internet



Fiona



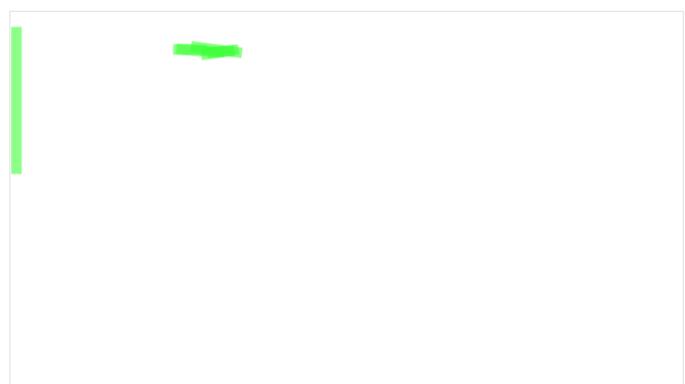
Kyo



Gabriela

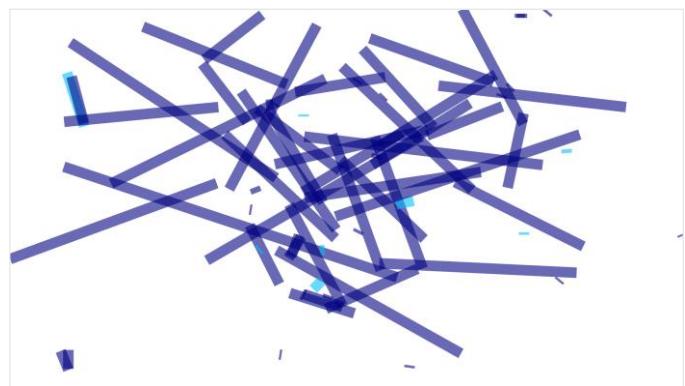


Simon



Julia

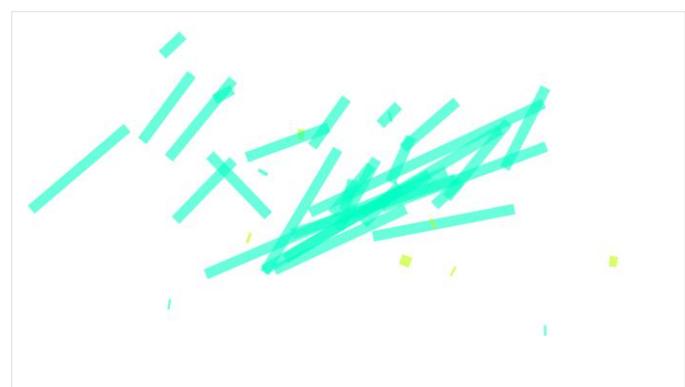
Andi



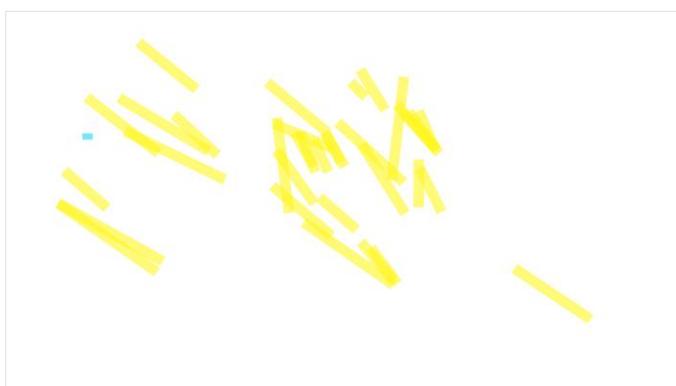
Fiona



Kyo



Gabriela



Simon

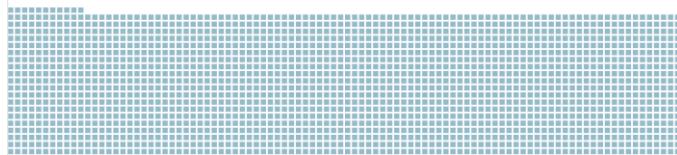
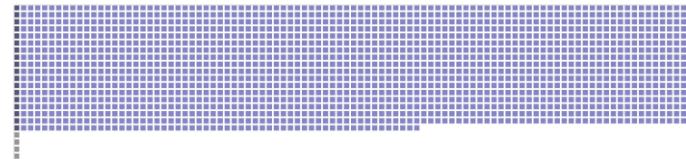
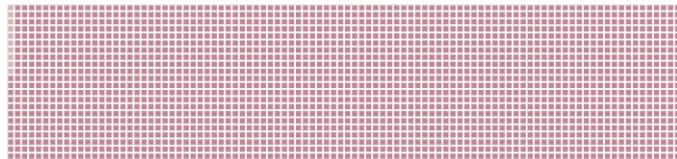


Julia

Andi

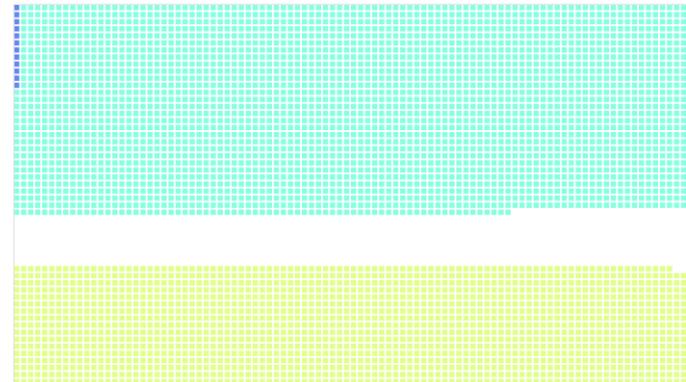
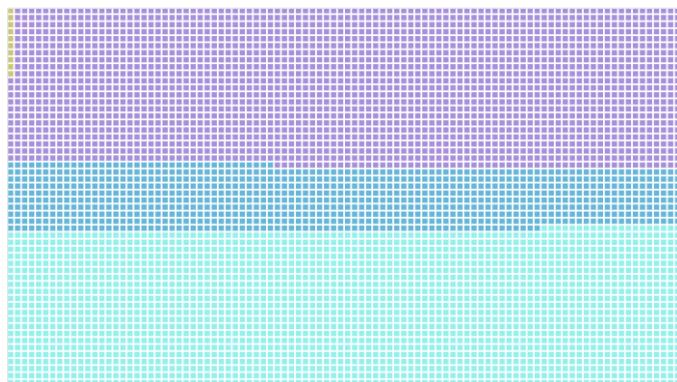
9.3.9 Programm 9

Aufsatz



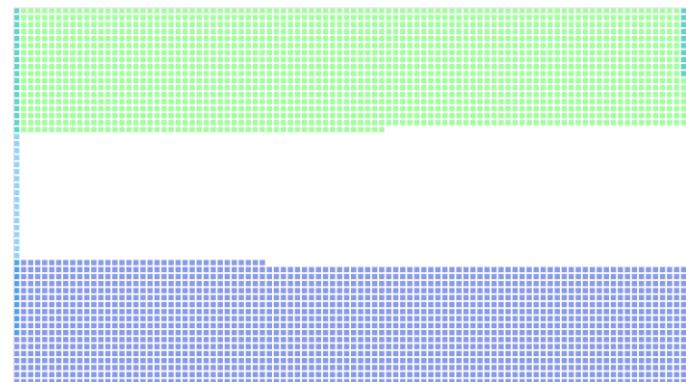
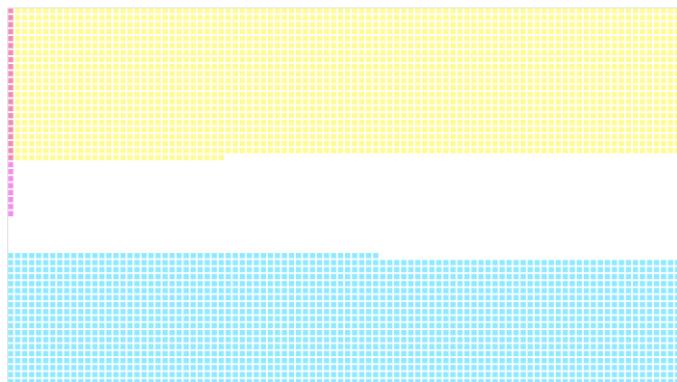
Fiona

Kyo



Gabriela

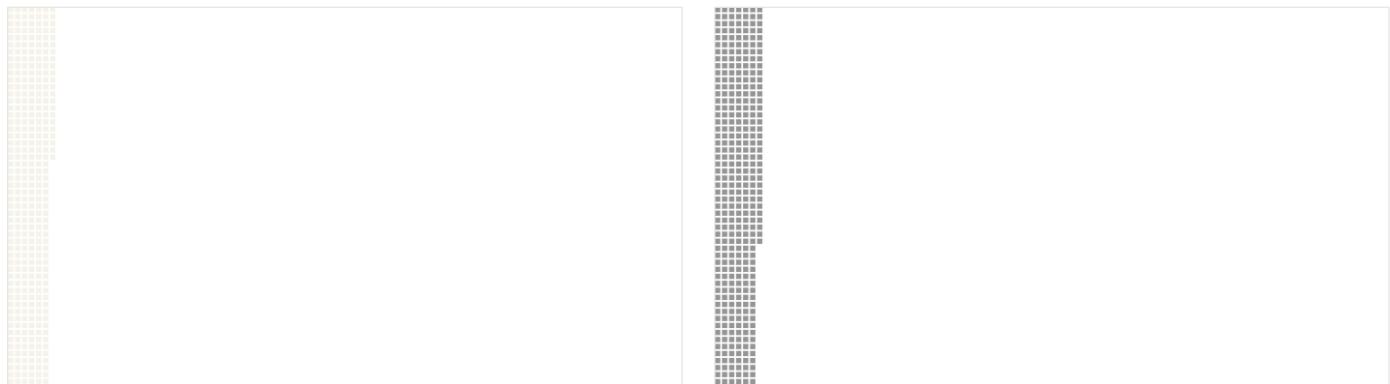
Simon



Julia

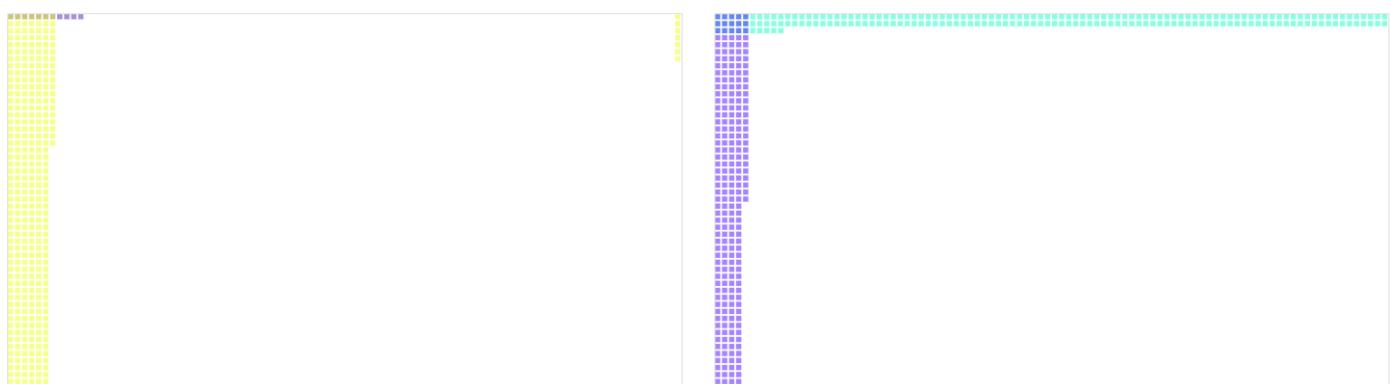
Andi

Bild



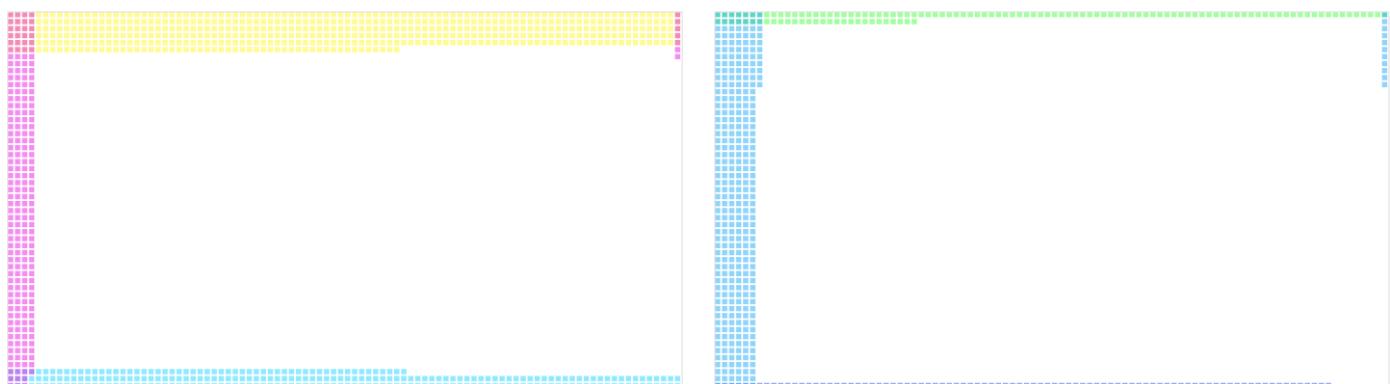
Fiona

Kyo



Gabriela

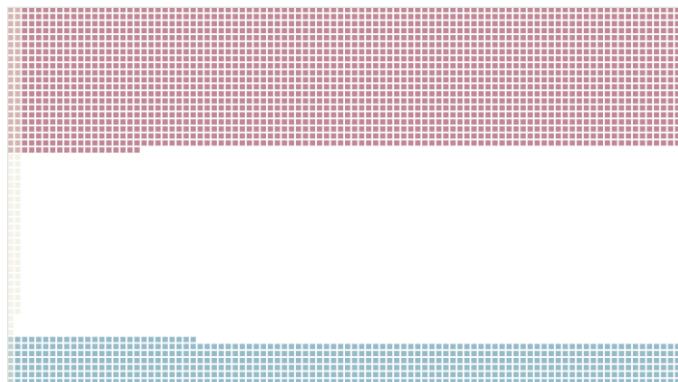
Simon



Julia

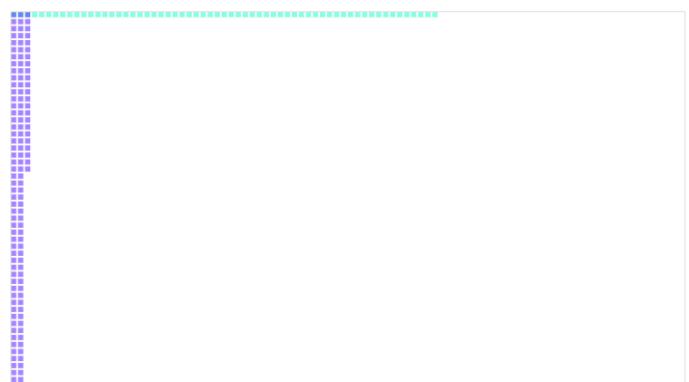
Andi

Internet



Fiona

Kyo



Gabriela

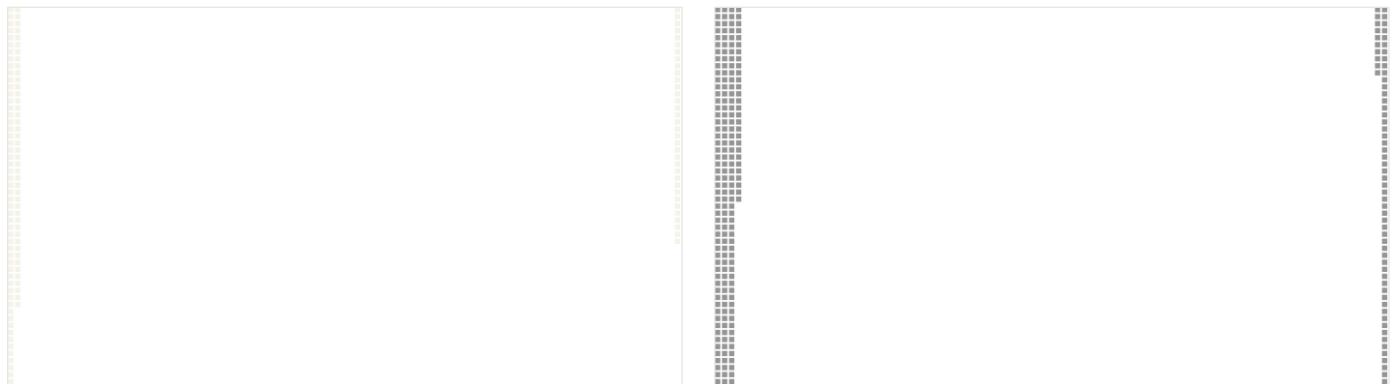
Simon



Julia

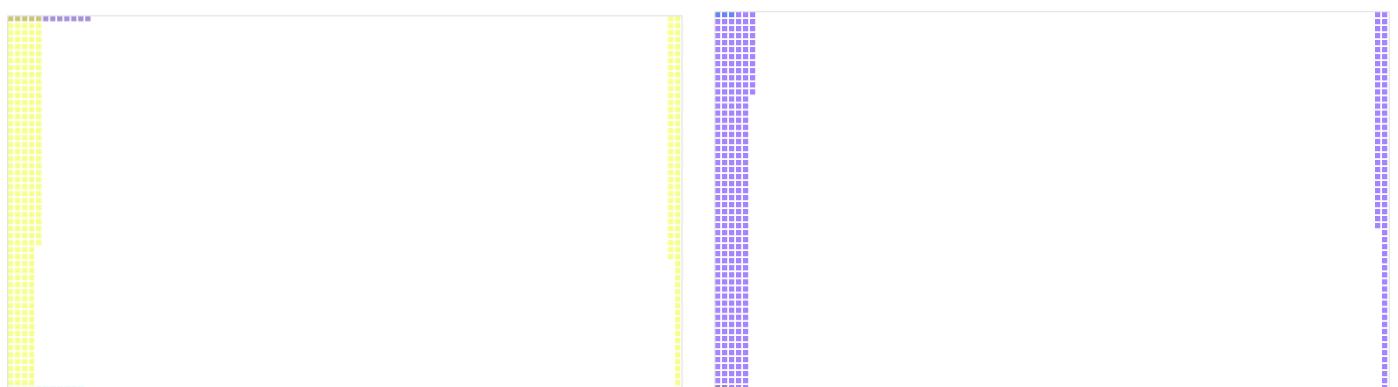
Andi

Spiel



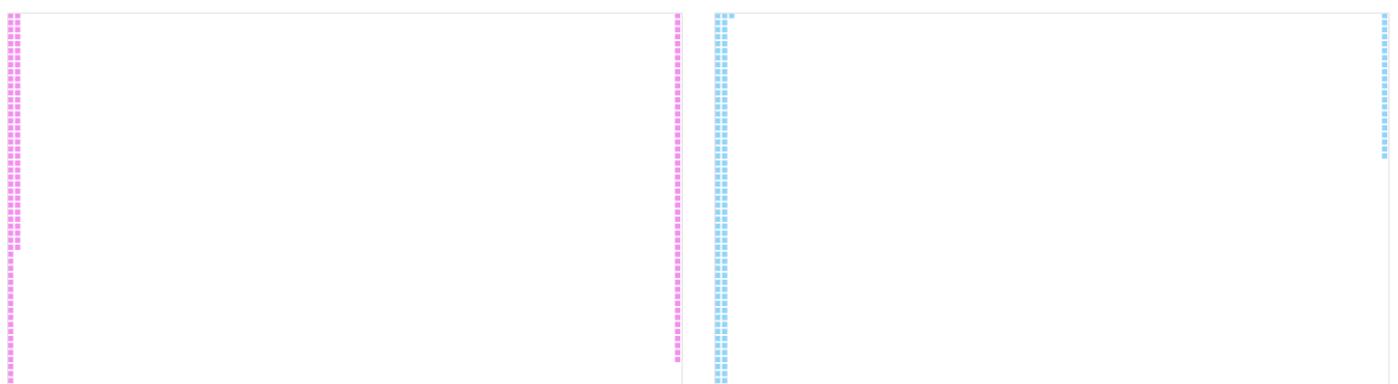
Fiona

Kyo



Gabriela

Simon

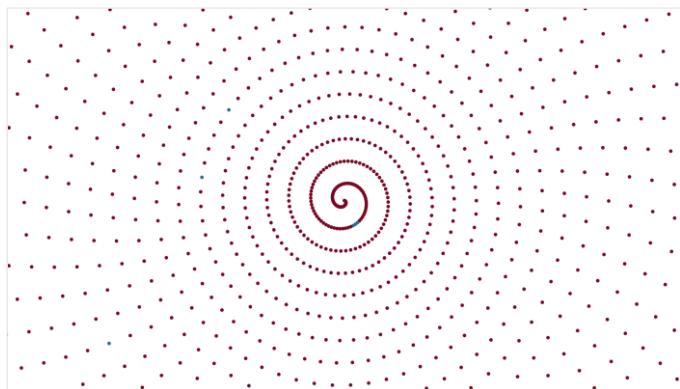


Julia

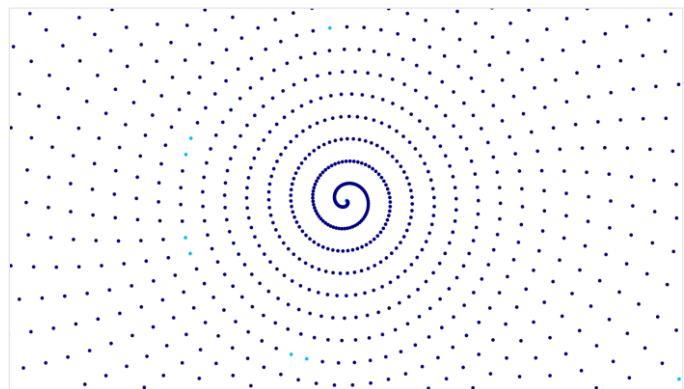
Andi

9.3.10 Programm 10

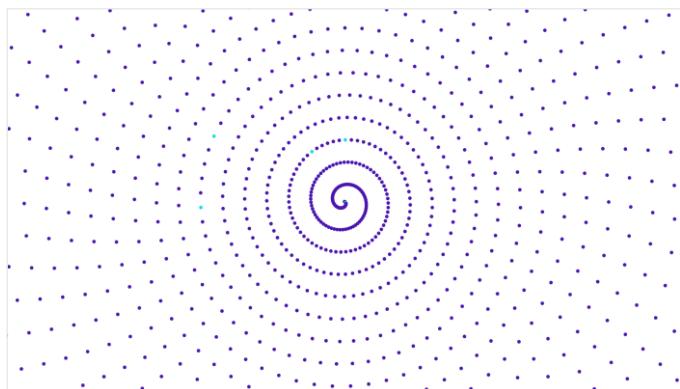
Aufsatz



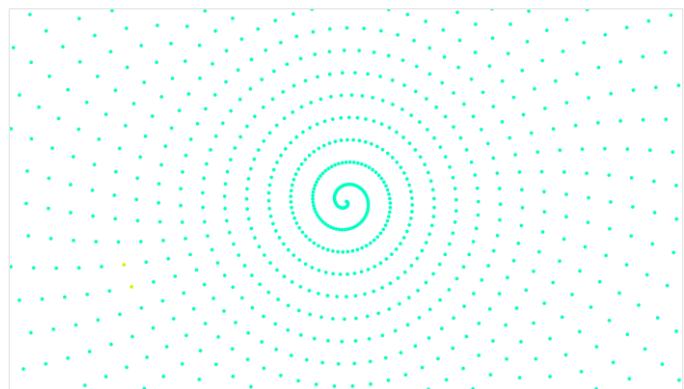
Fiona



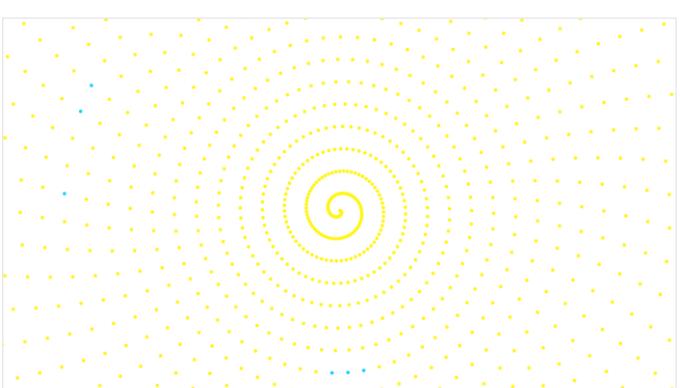
Kyo



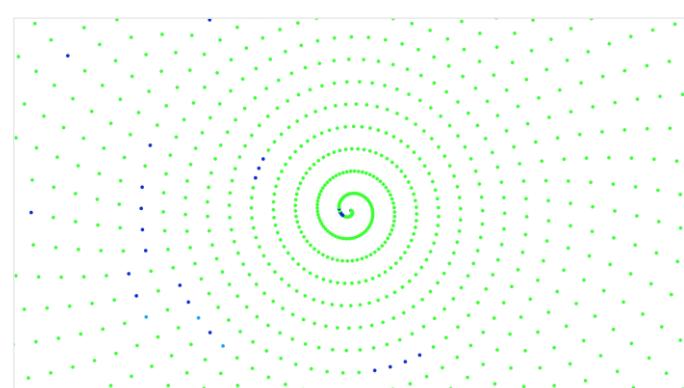
Gabriela



Simon



Julia



Andi

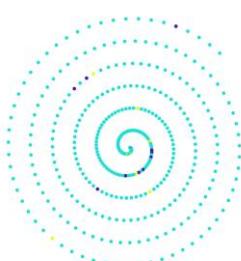
Bild



Fiona



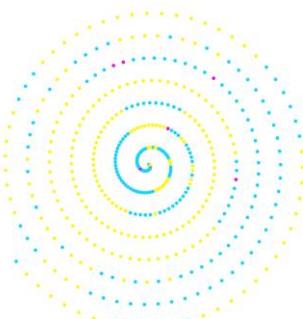
Kyo



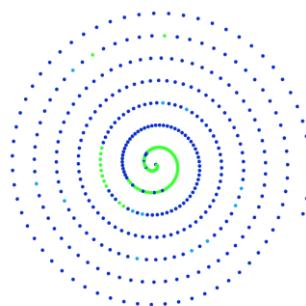
Gabriela



Simon

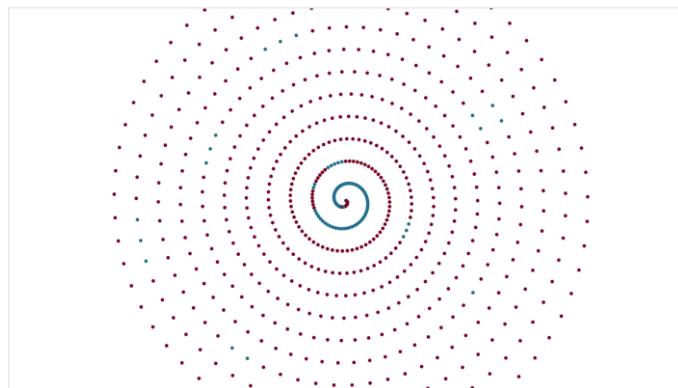


Julia

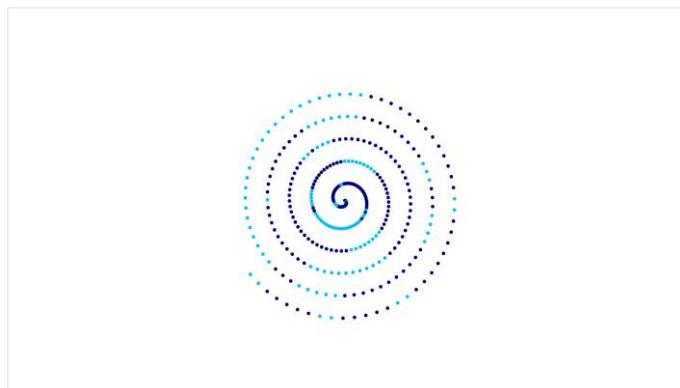


Andi

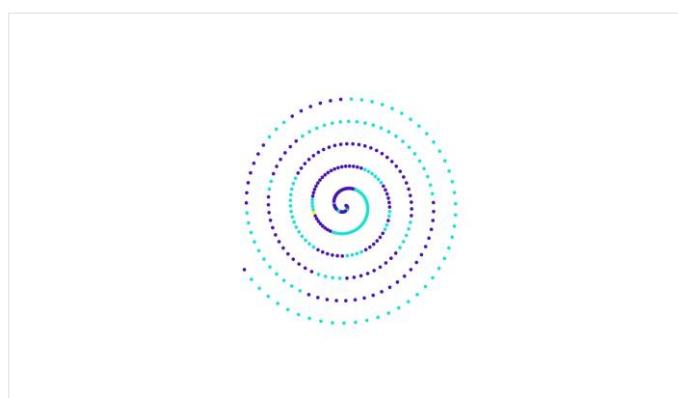
Internet



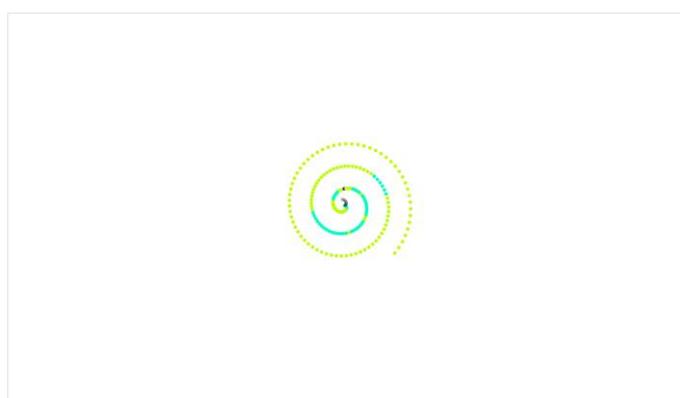
Fiona



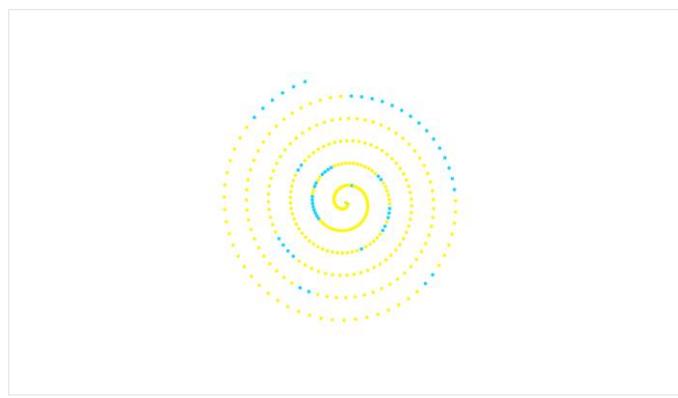
Kyo



Gabriela



Simon



Julia



Andi

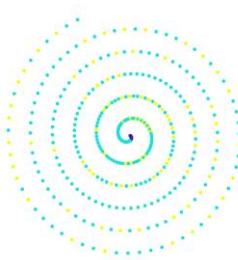
Spiel



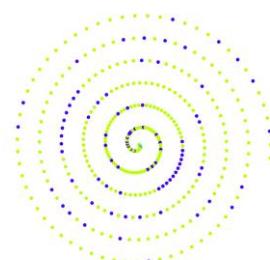
Fiona



Kyo



Gabriela



Simon



Julia



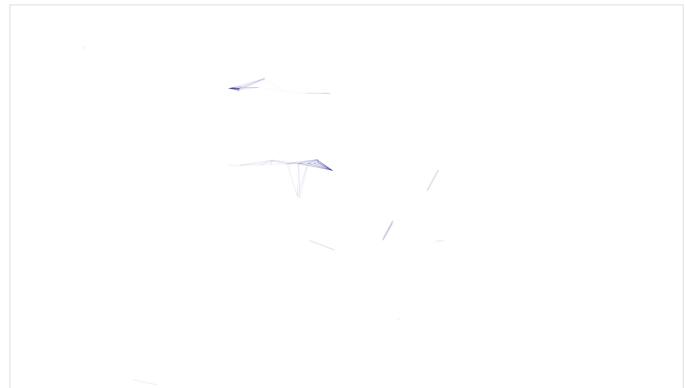
Andi

9.3.11 Programm 11

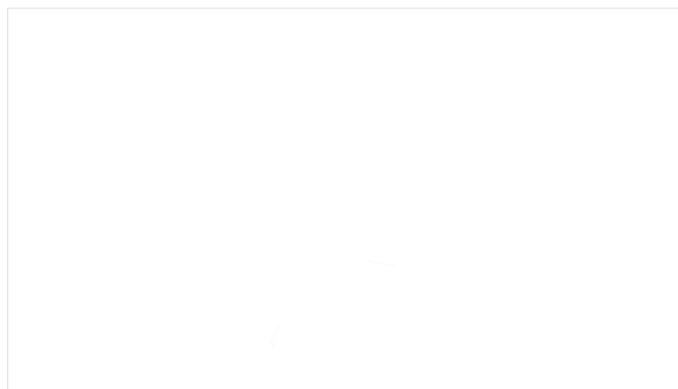
Aufsatz



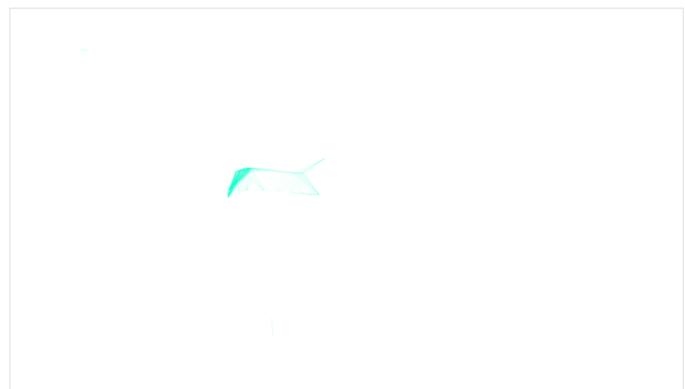
Fiona



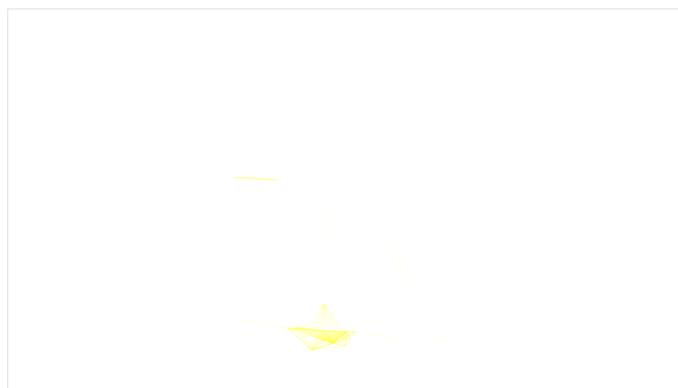
Kyo



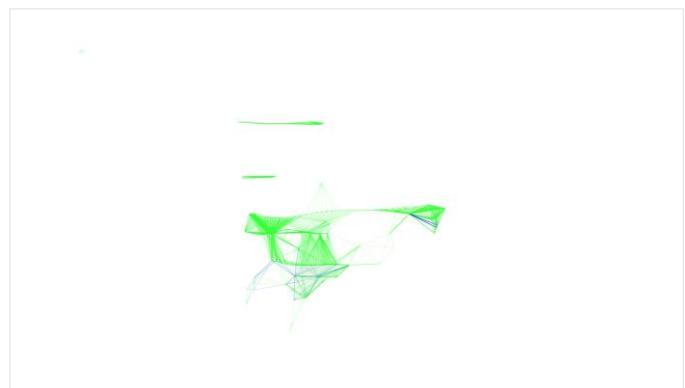
Gabriela



Simon

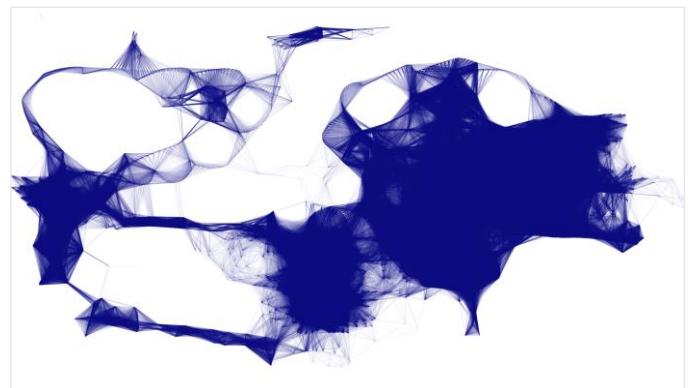


Julia

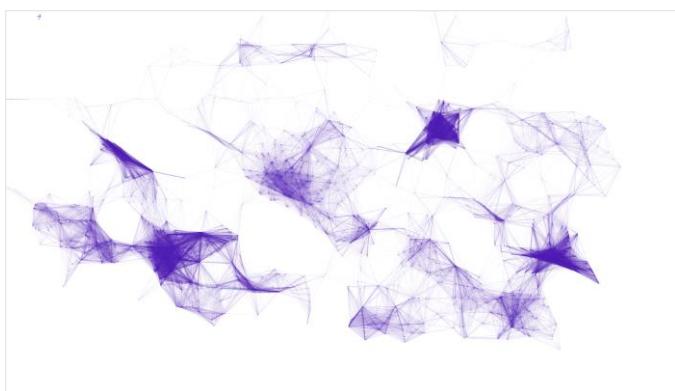


Andi

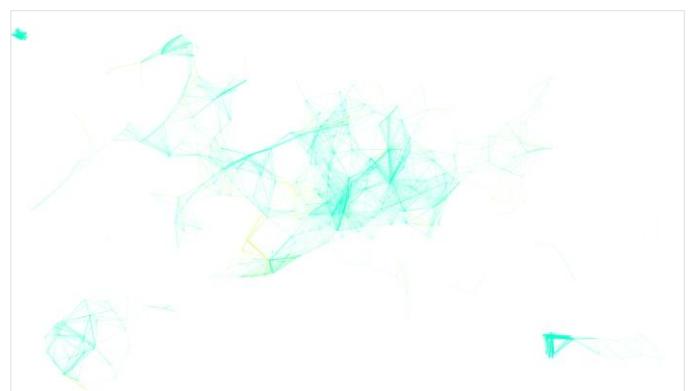
Bild



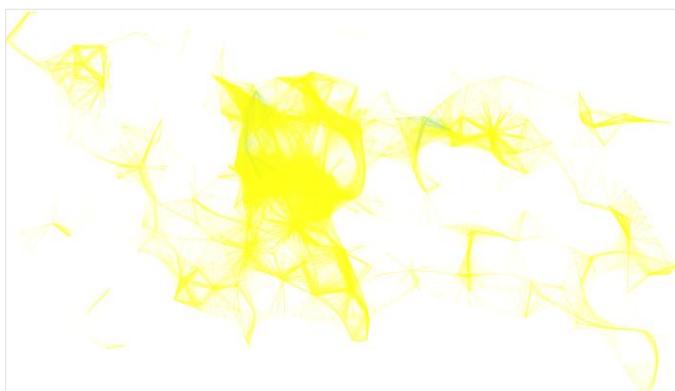
Fiona



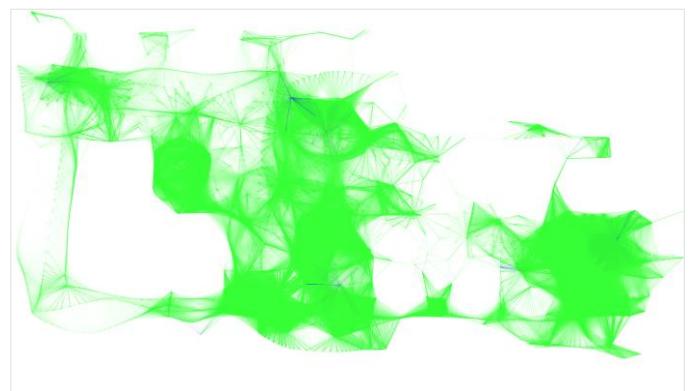
Kyo



Gabriela



Simon



Julia

Andi

Internet

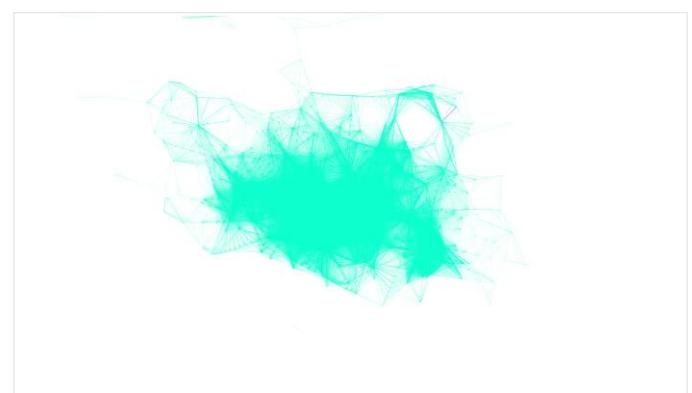


Fiona

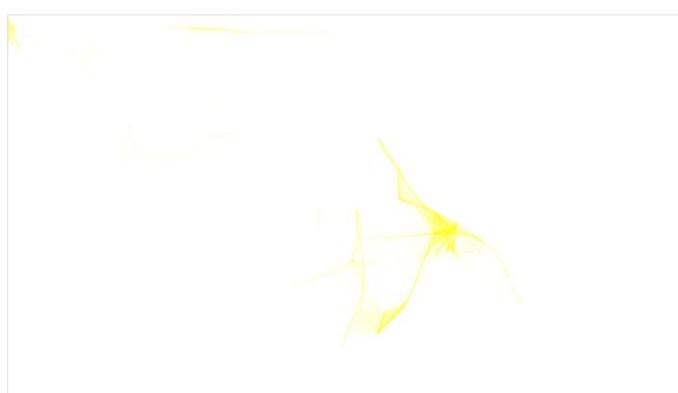
Kyo



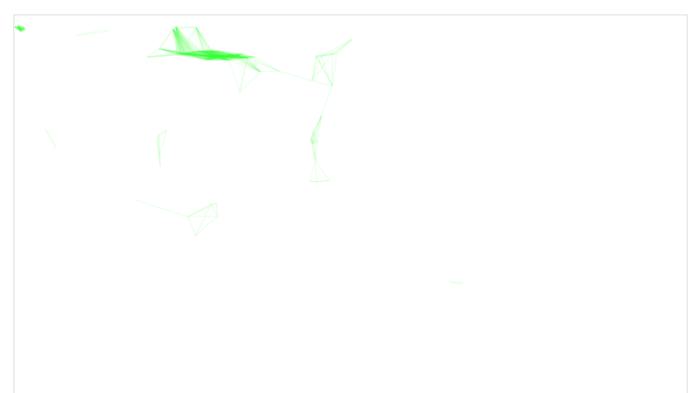
Gabriela



Simon

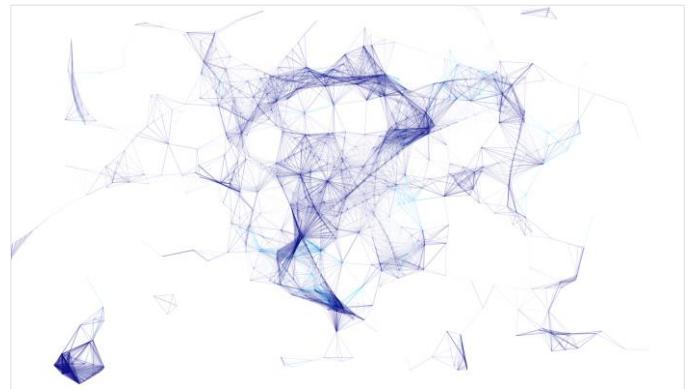


Julia

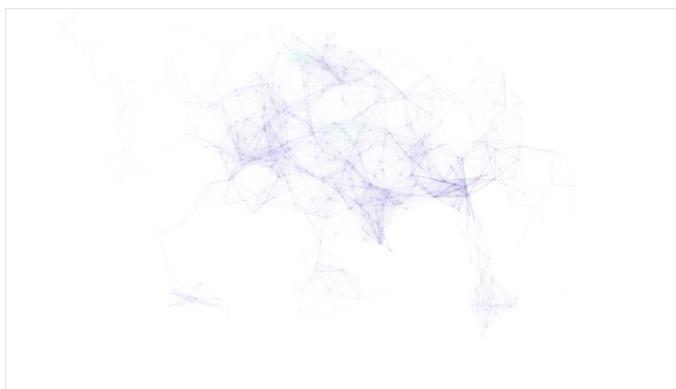


Andi

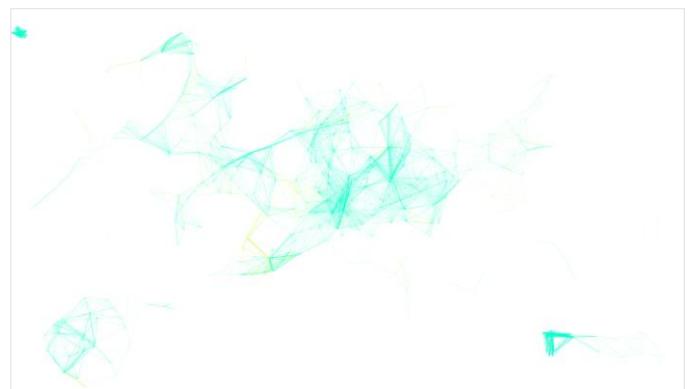
Spiel



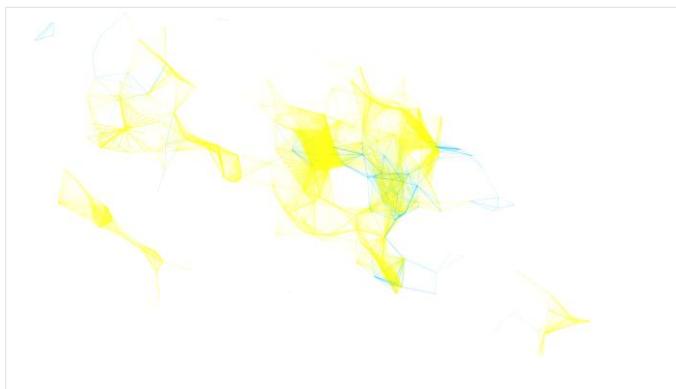
Fiona



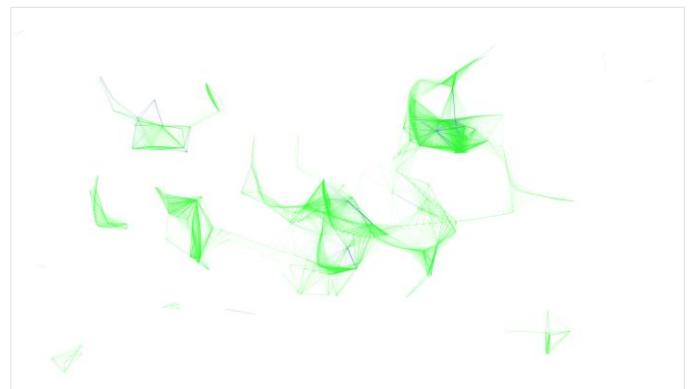
Kyo



Gabriela



Simon

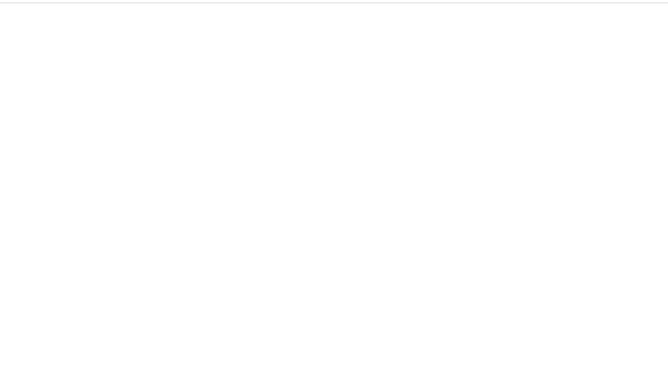


Julia

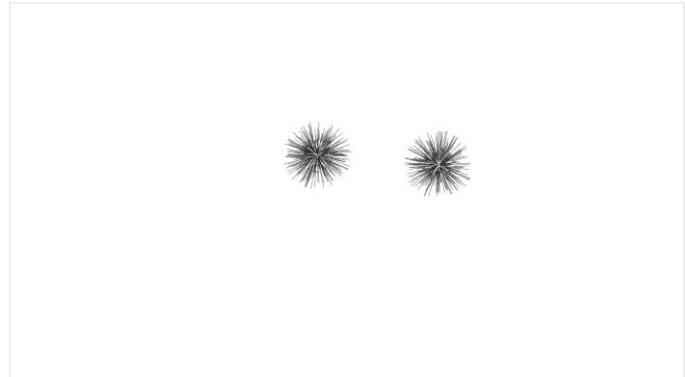
Andi

9.3.12 Programm 12

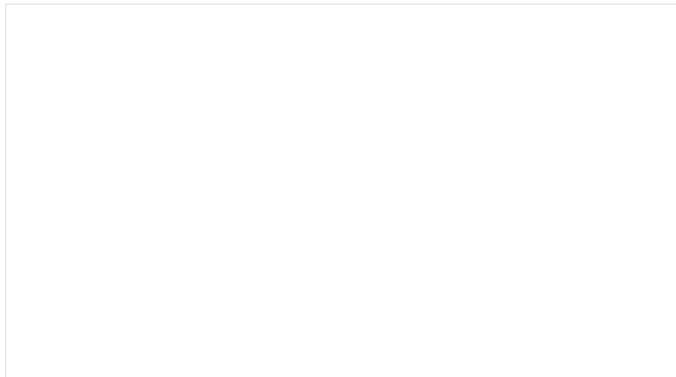
Aufsatz



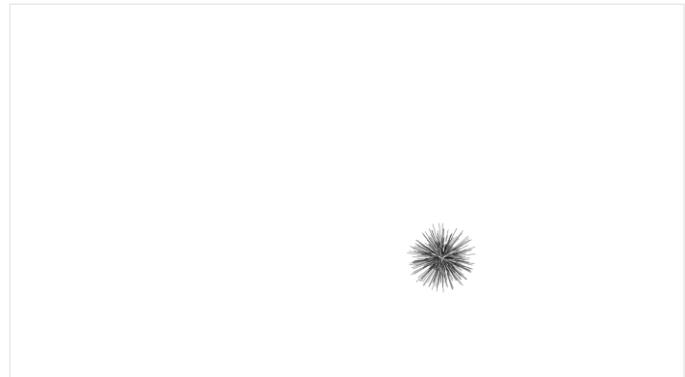
Fiona



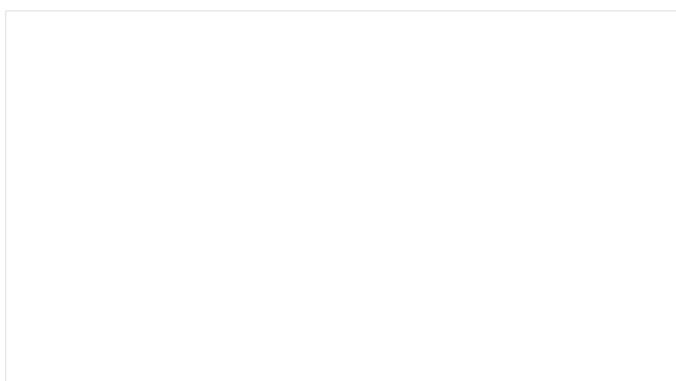
Kyo



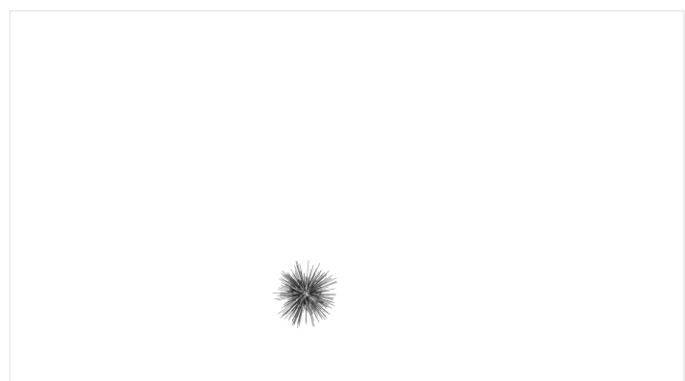
Gabriela



Simon

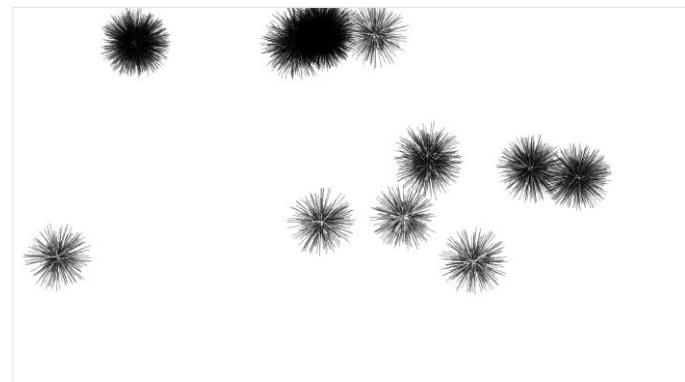
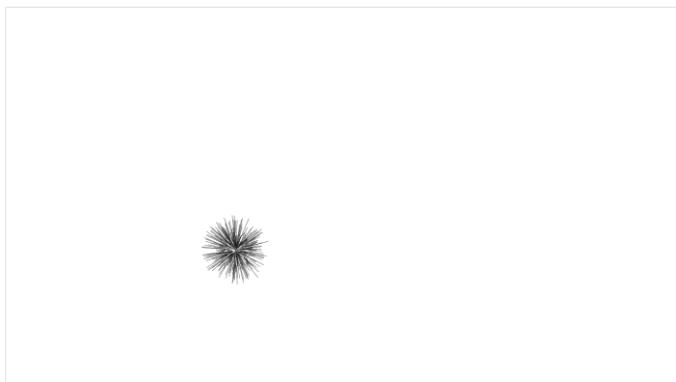


Julia

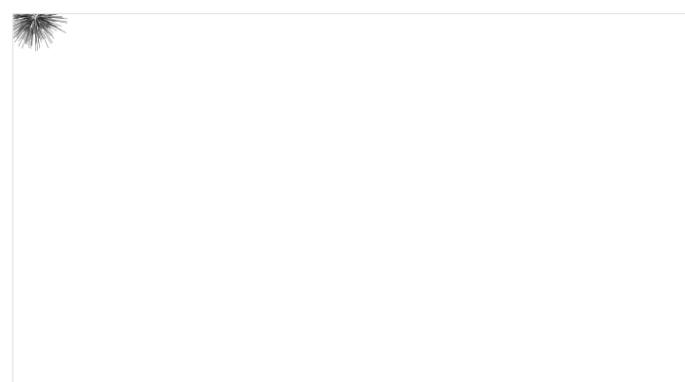
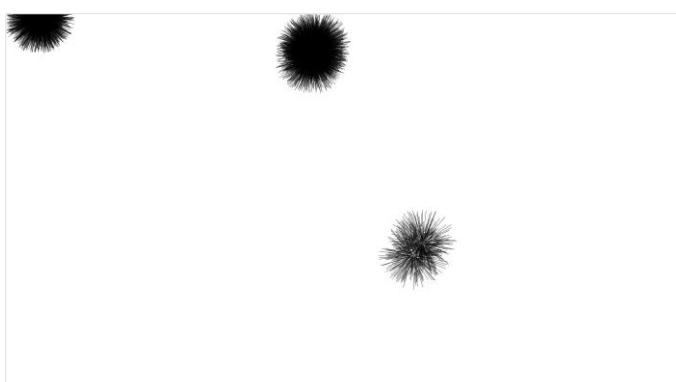


Andi

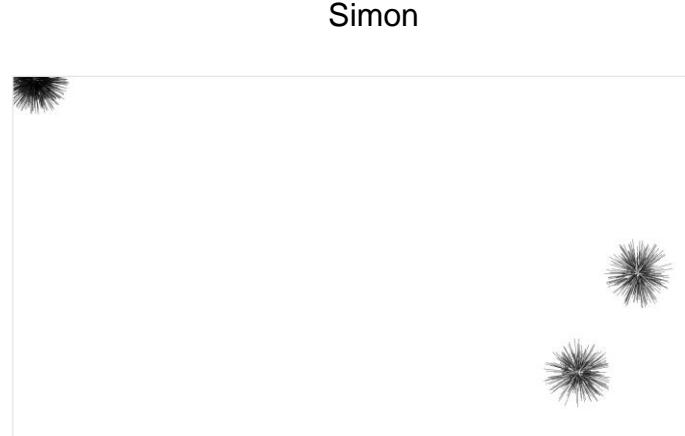
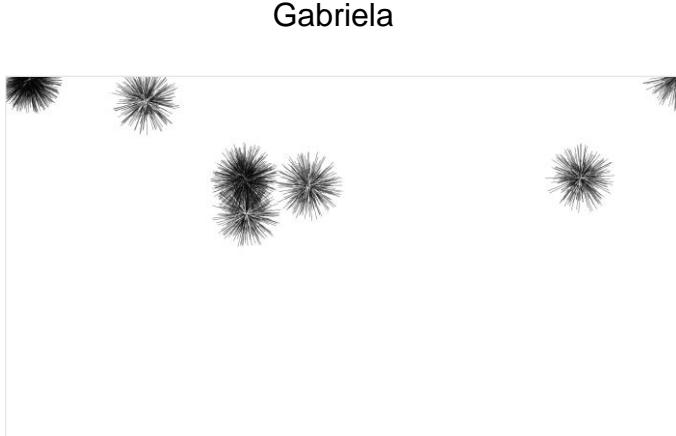
Bild



Fiona



Kyo



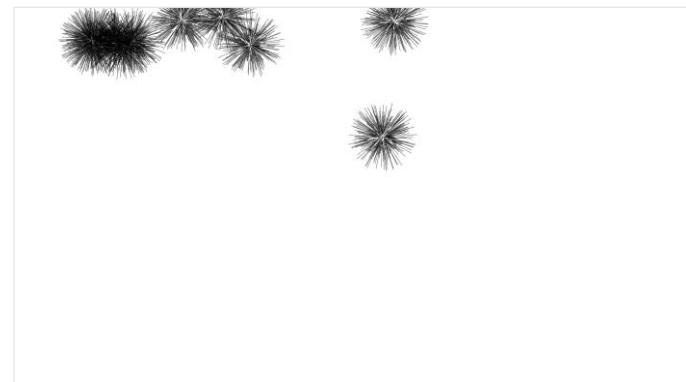
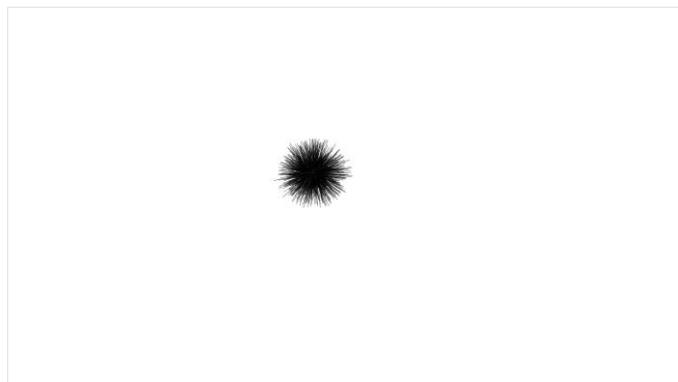
Gabriela

Simon

Julia

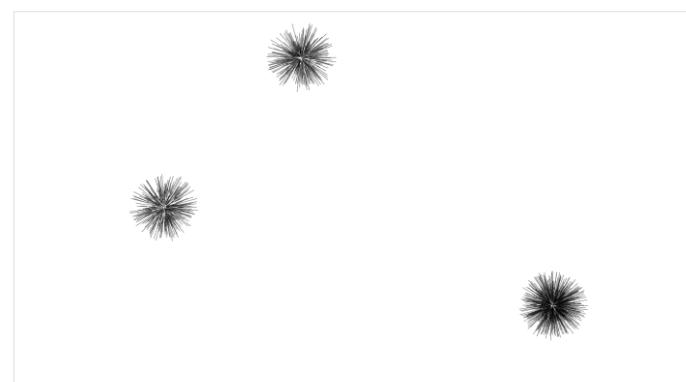
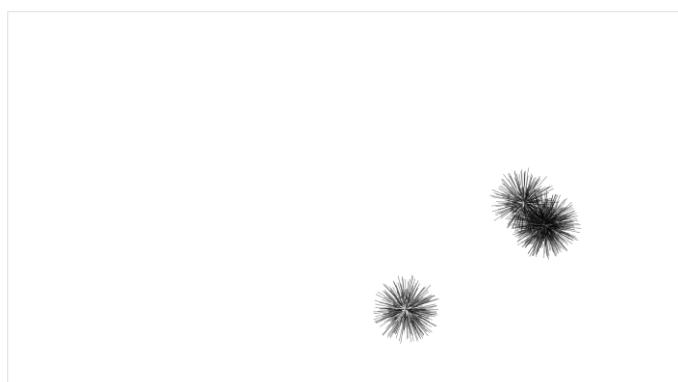
Andi

Internet



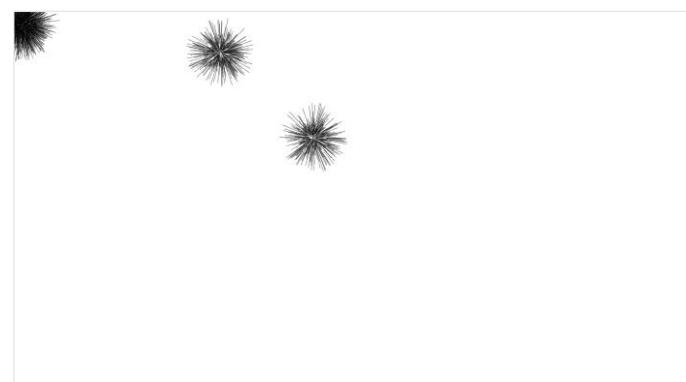
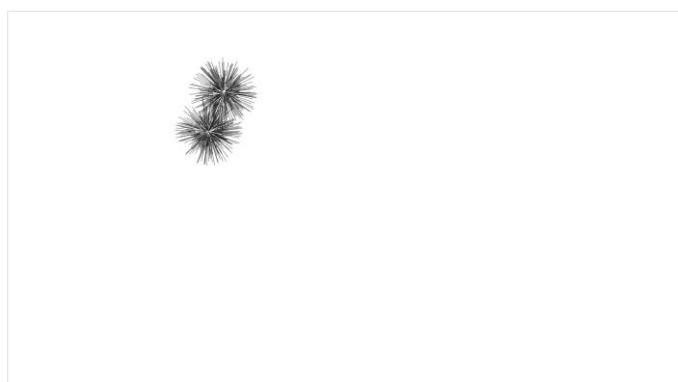
Fiona

Kyo



Gabriela

Simon



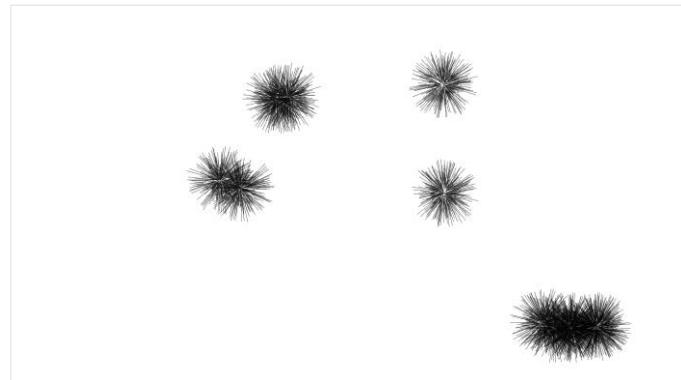
Julia

Andi

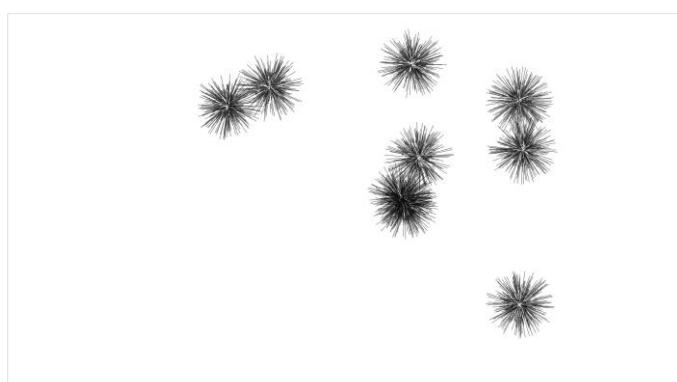
Spiel



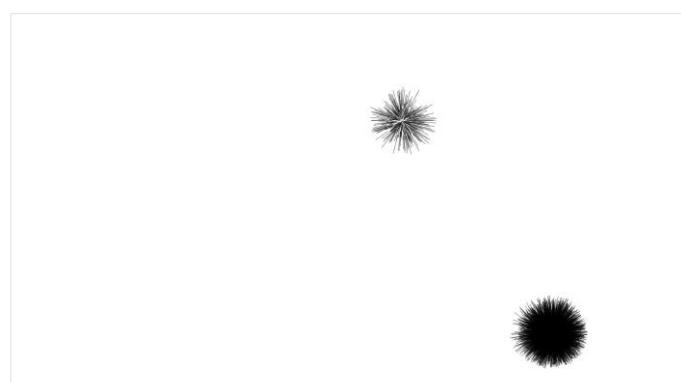
Fiona



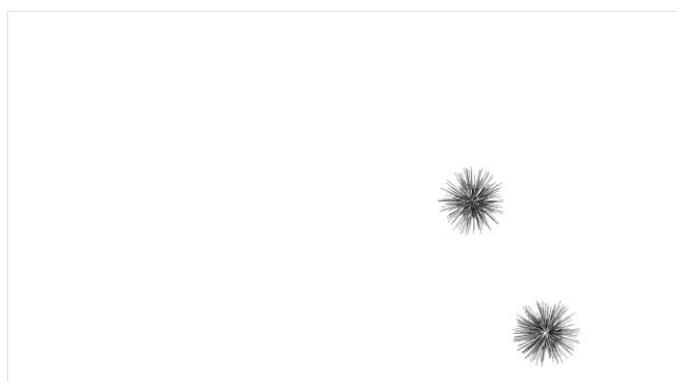
Kyo



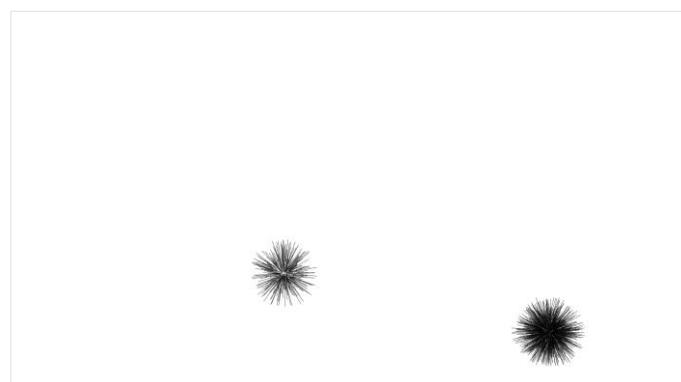
Gabriela



Simon



Julia

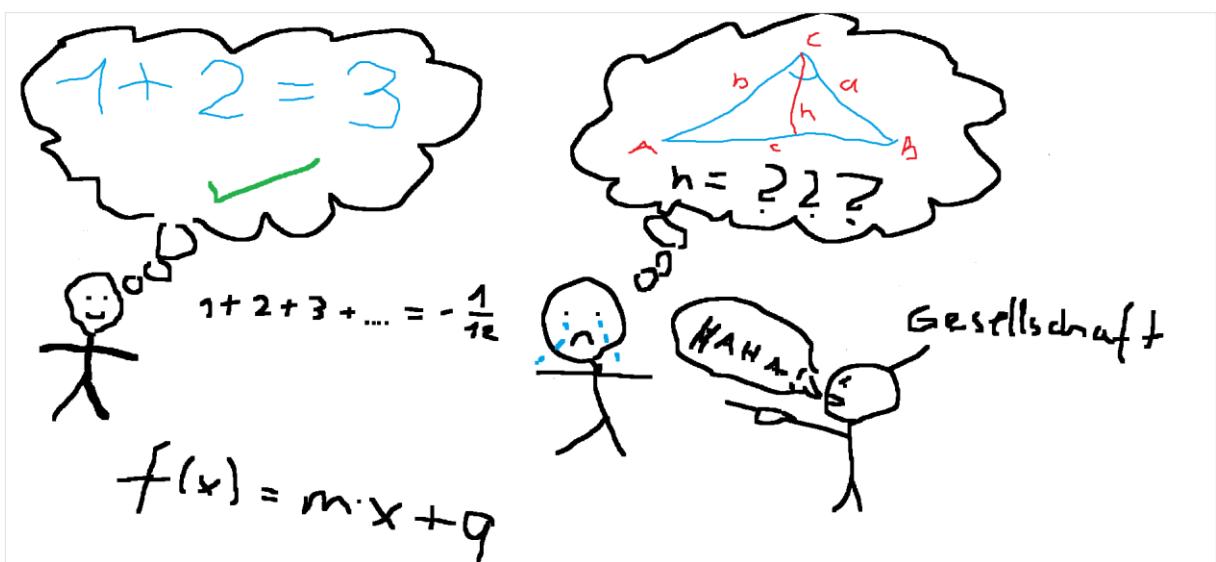


Andi

9.4 Von Testpersonen gemalten Bilder



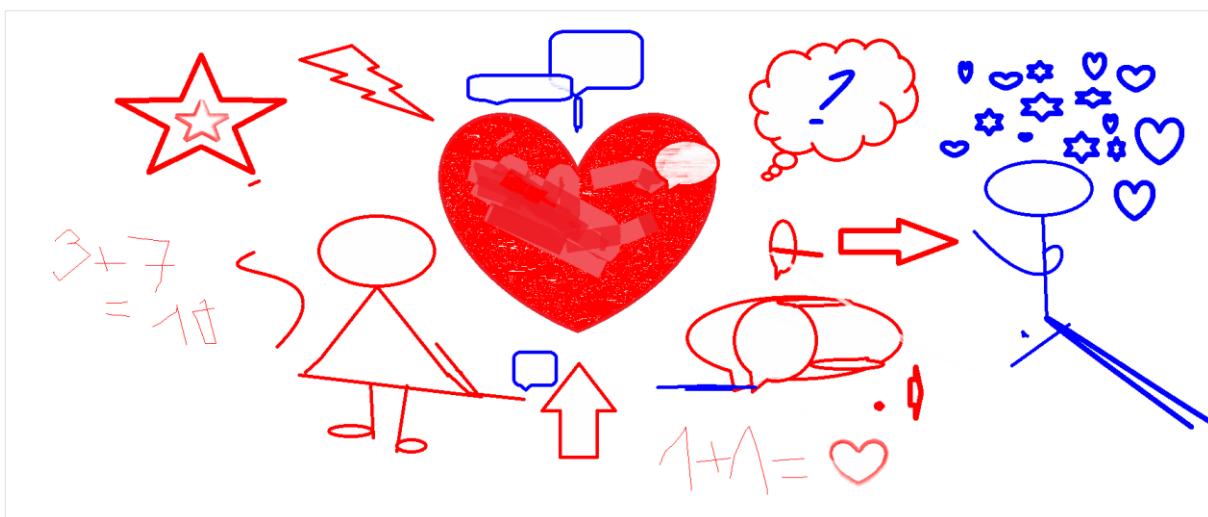
Fiona



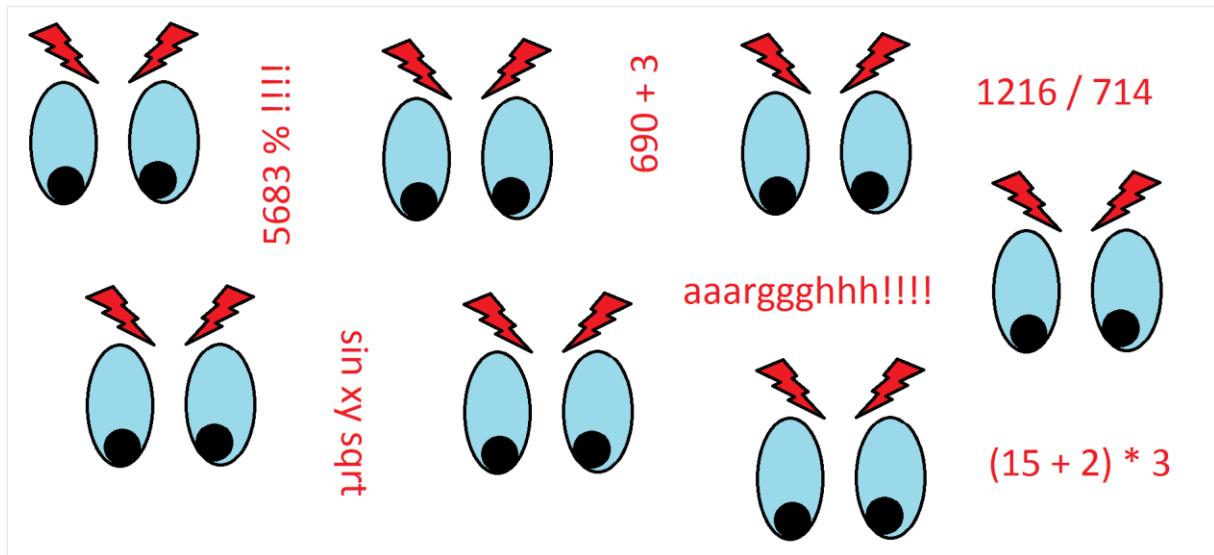
Kyo



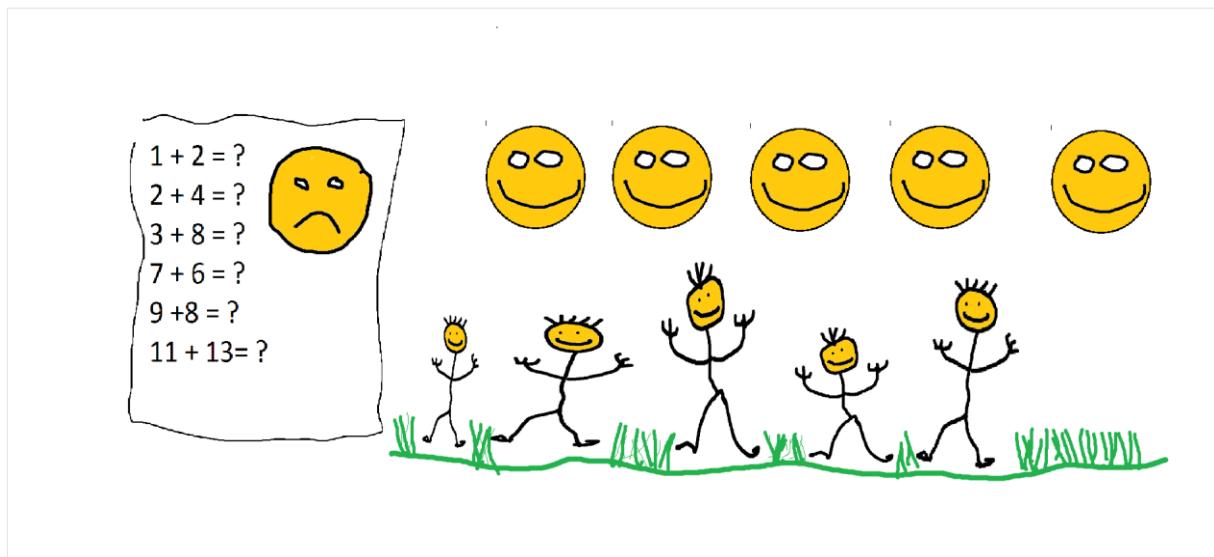
Simon



Gabriela



Julia



Andi

9.5 Angaben und Einverständniserklärung

Andi Hess Sonntag, 31. Juli 2016

Angaben und Einverständniserklärung

Vorname: Andi
Nachname: Hess
Alter: 55
Beruf: Leiter ICT Schulamt
Geschlecht:
 Mann Frau

Computerkenntnisse:
 sehr schlecht schlecht mittel gut sehr gut

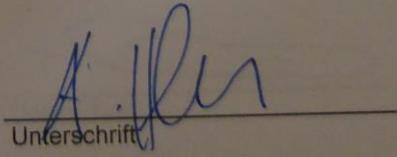
Farben (RGB Wert):
1.Farbe: 59,255,52
2.Farbe: 9,47,204
3.Farbe: 24,166,242

Ich bestätige, dass ich damit einverstanden bin, dass meine Angaben und erstellten Werke (Aufsatz und Bilder) im Rahmen der Maturarbeit 2016 von Neville Walo öffentlich publiziert werden können.

Ich wurde von der verantwortlichen Person vollständig über Wesen, Bedeutung und Tragweite der Studie aufgeklärt. Ich hatte die Möglichkeit, Fragen zu stellen. Ich habe die Antworten verstanden und akzeptiere sie. Ich bin über die mit der Teilnahme an der Studie verbundenen Risiken und auch über den möglichen Nutzen informiert. Ich hatte ausreichend Zeit, mich zur Teilnahme an der Studie zu entscheiden und weiß, dass die Teilnahme freiwillig ist. Ich wurde darüber informiert, dass ich jederzeit und ohne Angabe von Gründen diese Zustimmung widerrufen kann, ohne dass dadurch Nachteile für mich entstehen.

Ich erkläre hiermit meine freiwillige Teilnahme an dieser Studie.

31.7.2016
Ort, Datum


Unterschrift

Seite 1

Julia

Sonntag, 31. Juli 2016

Angaben und Einverständniserklärung

Vorname: Julia

Nachname: Rennenkampff

Alter: 52

Beruf: Mathematikerin, Psychologin, Schulleiterin

Geschlecht:

Mann Frau

Computerkenntnisse:

sehr schlecht schlecht mittel gut sehr gut

Farben (RGB Wert):

1. Farbe: 255,246,23

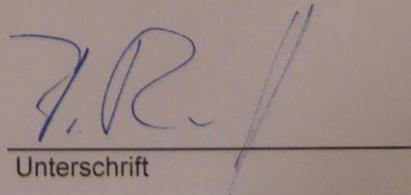
2. Farbe: 22,212,255

3. Farbe: 232,17,224

- Ich bestätige, dass ich damit einverstanden bin, dass meine Angaben und erstellten Werke (Aufsatz und Bilder) im Rahmen der Maturarbeit 2016 von Neville Walo öffentlich publiziert werden können.
- Ich wurde von der verantwortlichen Person vollständig über Wesen, Bedeutung und Tragweite der Studie aufgeklärt. Ich hatte die Möglichkeit, Fragen zu stellen. Ich habe die Antworten verstanden und akzeptiere sie. Ich bin über die mit der Teilnahme an der Studie verbundenen Risiken und auch über den möglichen Nutzen informiert. Ich hatte ausreichend Zeit, mich zur Teilnahme an der Studie zu entscheiden und weiß, dass die Teilnahme freiwillig ist. Ich wurde darüber informiert, dass ich jederzeit und ohne Angabe von Gründen diese Zustimmung widerrufen kann, ohne dass dadurch Nachteile für mich entstehen.
- Ich erkläre hiermit meine freiwillige Teilnahme an dieser Studie.

Zürich, 31.7.16

Ort, Datum


Unterschrift

Seite 1

Gabriela

Mittwoch, 10. August 2016

Angaben und Einverständniserklärung

Vorname: Gabriela

Nachname: Rosser

Alter: 25

Beruf: Lehrerin

Geschlecht:

Mann Frau

Computerkenntnisse:

sehr schlecht schlecht mittel gut sehr gut

Farben (RGB Wert):

1.Farbe:74,21,175

2.Farbe:29,224,213

3.Farbe:238,255,13

- Ich bestätige, dass ich damit einverstanden bin, dass meine Angaben und erstellten Werke (Aufsatz und Bilder) im Rahmen der Maturarbeit 2016 von Neville Walo öffentlich publiziert werden können.
- Ich wurde von der verantwortlichen Person vollständig über Wesen, Bedeutung und Tragweite der Studie aufgeklärt. Ich hatte die Möglichkeit, Fragen zu stellen. Ich habe die Antworten verstanden und akzeptiere sie. Ich bin über die mit der Teilnahme an der Studie verbundenen Risiken und auch über den möglichen Nutzen informiert. Ich hatte ausreichend Zeit, mich zur Teilnahme an der Studie zu entscheiden und weiß, dass die Teilnahme freiwillig ist. Ich wurde darüber informiert, dass ich jederzeit und ohne Angabe von Gründen diese Zustimmung widerrufen kann, ohne dass dadurch Nachteile für mich entstehen.
- Ich erkläre hiermit meine freiwillige Teilnahme an dieser Studie.



Flumserberg 9.8.2016

Ort, Datum

Unterschrift

Seite 1

Kyo Mangold

Angaben und Einverständniserklärung

Vorname: Kyo
Nachname: Mangold
Alter: 18
Beruf: Schüler
Geschlecht:
 Mann Frau

Computerkenntnisse:
 sehr schlecht schlecht mittel gut sehr gut

Farben (RGB Wert):
1. Farbe: 0,0,128
2. Farbe: 0,191,255
3. Farbe: 33,33,33

Ich bestätige, dass ich damit einverstanden bin, dass meine Angaben und erstellten Werke (Aufsatz und Bilder) im Rahmen der Maturarbeit 2016 von Neville Walo öffentlich publiziert werden können.

Ich wurde von der verantwortlichen Person vollständig über Wesen, Bedeutung und Tragweite der Studie aufgeklärt. Ich hatte die Möglichkeit, Fragen zu stellen. Ich habe die Antworten verstanden und akzeptiere sie. Ich bin über die mit der Teilnahme an der Studie verbundenen Risiken und auch über den möglichen Nutzen informiert. Ich hatte ausreichend Zeit, mich zur Teilnahme an der Studie zu entscheiden und weiß, dass die Teilnahme freiwillig ist. Ich wurde darüber informiert, dass ich jederzeit und ohne Angabe von Gründen diese Zustimmung widerrufen kann, ohne dass dadurch Nachteile für mich entstehen.

Ich erkläre hiermit meine freiwillige Teilnahme an dieser Studie.

Höngg, 12.7.2016

Unterschrift

Simon Walo Montag, 25. Juli 2016

Angaben und Einverständniserklärung

Vorname: Simon
Nachname: Walo
Alter: 24
Beruf: Student
Geschlecht:
 Mann Frau

Computerkenntnisse:
 sehr schlecht schlecht mittel gut sehr gut

Farben (RGB Wert):
1. Farbe: 0,255,195
2. Farbe: 188,255,0
3. Farbe: 66,0,255

Ich bestätige, dass ich damit einverstanden bin, dass meine Angaben und erstellten Werke (Aufsatz und Bilder) im Rahmen der Maturarbeit 2016 von Neville Walo öffentlich publiziert werden können.

Ich wurde von der verantwortlichen Person vollständig über Wesen, Bedeutung und Tragweite der Studie aufgeklärt. Ich hatte die Möglichkeit, Fragen zu stellen. Ich habe die Antworten verstanden und akzeptiere sie. Ich bin über die mit der Teilnahme an der Studie verbundenen Risiken und auch über den möglichen Nutzen informiert. Ich hatte ausreichend Zeit, mich zur Teilnahme an der Studie zu entscheiden und weiß, dass die Teilnahme freiwillig ist. Ich wurde darüber informiert, dass ich jederzeit und ohne Angabe von Gründen diese Zustimmung widerrufen kann, ohne dass dadurch Nachteile für mich entstehen.

Ich erkläre hiermit meine freiwillige Teilnahme an dieser Studie.

Zürich, 25.7.16
Ort, Datum


Unterschrift

Seite 1

Fiona Seitz

Sonntag, 10. Juli 2016

Angaben und Einverständniserklärung

Vorname: Fiona

Nachname: Seitz

Alter: 17 Jahre

Beruf: Schülerin

Geschlecht:

 Mann Frau

Computerkenntnisse:

 sehr schlecht schlecht mittel gut sehr gut

Farben (RGB Wert):

1. Farbe: 121,6,35

2. Farbe: 33,112,142

3. Farbe: 234,229,206

- Ich bestätige, dass ich damit einverstanden bin, dass meine Angaben und erstellten Werke (Aufsatz und Bilder) im Rahmen der Maturarbeit 2016 von Neville Walo öffentlich publiziert werden können.
- Ich wurde von der verantwortlichen Person vollständig über Wesen, Bedeutung und Tragweite der Studie aufgeklärt. Ich hatte die Möglichkeit, Fragen zu stellen. Ich habe die Antworten verstanden und akzeptiere sie. Ich bin über die mit der Teilnahme an der Studie verbundenen Risiken und auch über den möglichen Nutzen informiert. Ich hatte ausreichend Zeit, mich zur Teilnahme an der Studie zu entscheiden und weiß, dass die Teilnahme freiwillig ist. Ich wurde darüber informiert, dass ich jederzeit und ohne Angabe von Gründen diese Zustimmung widerrufen kann, ohne dass dadurch Nachteile für mich entstehen.
- Ich erkläre hiermit meine freiwillige Teilnahme an dieser Studie.

Höngg, 10. 7. 16
Ort, Datum

F. S.
Unterschrift

9.6 Aufsätze

Andi Hess

Sonntag, 31. Juli 2016

Aufsatz

Schreibe einen Aufsatz aus den folgenden Sätzen und Wörtern. Setze einen eigenen Titel.

während er ging, ging etwas kaputt

lächelt

Der Computer

55

Der Crash

Es war ein angenehmer Sommerabend, Jules sass im Garten und trank ein Bier. Während er am Glas nippte, ging er im Kopf die vergangenen Monate durch und lächelte. Immer wieder musste er an die Begegnungen mit Mona denken, an die innigen Küsse und vieles mehr. Seit 55 Wochen waren sie nun zusammen und hatten die gemeinsame Zeit genossen. Bis auf jeden Moment, als der Computer streikte. Wirklich im dümmsten Augenblick stieg die alte Kiste aus und verursachte ein Chaos, denn Mona war gerade daran eine wichtige Arbeit abzuschliessen. Sie war daran, das letzte Kapitel zu schreiben, als der Rechner plötzlich begann, seltsam zu riechen. Jules musste zu einem wichtigen Meeting, zu dem er rechtzeitig erscheinen musste. Er war schon daran das Haus zu verlassen, als es passiert. Genau während er ging, ging etwas kaputt auf der Hauptplatine und der Computer verabschiedete sich mit einem leisen Knall. Mona fluchte, denn sie hatte das letzte Kapitel noch nicht gespeichert. Und sie fluchte noch viel mehr, als sich der Rechner nicht mehr starten liess. Denn jetzt wurde ihr bewusst, dass sie keinen Zugriff mehr auf ihr wichtiges Dokument hatte. Jules nahm noch eine Schluck Bier und erinnerte sich daran, wie er die Harddisk ausbaute und die Daten rettete. Leider verpasste er sein Meeting und verlor deshalb seinen Job.

Gabriela

Mittwoch, 10. August 2016

Aufsatz

Schreibe einen Aufsatz aus den folgenden Sätzen und Wörtern. Setze einen eigenen Titel.

Heute war ich mit meinem Bruder in den Bergen. Leider musst er seinen Computer mitbringen, da er bald seine Semesterprüfungen schreiben musste. Trotzdem war es schön, ihn bei diesem Wetter dabei zu haben. Wir haben viel gelacht. In der Mittagspause wanderten wir zu einer wunderbar, alten Berghütte und wir mussten beide grinsen, als uns die Hütte von unserer Grossmutter dabei in den Sinn kam. Als sie 55 Jahre alt war, kaufte sie sich mit unserem Grossvater zusammen eine Berghütte und die beiden verbrachten dort ihren Lebensabend gemeinsam. Sie erlebten viele Lustige Dinge wie zum Beispiel Eichhörnchen im Schlafzimmer und Wildschweine, die sie am Morgen jeweils begrüßten.

Während mein Bruder mit seinem Laptop in der Hand ging, ging etwas kaputt. Ich begriff zuerst gar nicht was es war, jedoch fuchtelte er wie wild um sich und fluchte mit lauter Stimme. Als ich begriff was es war, sah ich ihn bereits am Boden sitzen und seinen Computer liebkosend. War der Computer kaputt? Was war geschehen? Ich wusste, dass mein Bruder darauf sein ganzes Leben abgespeichert hatte. Ich schickte ein kurzes Stossgebet zum Himmel, denn ein kaputter Computer konnte in diesem Moment niemand gebrauchen. Zum Glück erhelltet sich sein Gesicht kurze Zeit später und ein erleichtertes Lächeln gab mir Zuversicht, dass nichts Schlimmes passiert war.

Dieser Schreckmoment liess meinen Bruder folgenden Entschluss fassen: Gelernt und am Computer gesessen werde in Zukunft zu Hause und von nun an wolle er den wunderbaren Tag ohne dieses Gerät verbringen. Ich lächelte ihn an und war froh meinen Bruder nun ganz für mich alleine zu haben. Wir wanderten unbeschwert weiter und fanden die Hütte, nach welcher wir Ausschau hielten etwa eine Stunde später.

Wiederrum mussten wir uns an unsere Grosseltern erinnern, als wir auf der Menu-Karte Omas – Apfelstrudel entdeckten. Dies wurde unsere Stärkung für den Weg. Bald schon mussten wir weiterziehen, denn wir sahen ein Gewitter aufkommen. Mein Bruder sah mich etwas besorgt an und fragte, ob es wohl besser sei wieder zurück ins Tal zu kehren. Mein Kopf liess es nicht zu die Wanderung ganz nach oben zum Gipfel einfach aufzugeben. Ich wollte es unbedingt schaffen. Mit etwas mulmigem Gefühl machten wir uns trotzdem auf in Richtung Höchster Punkt unserer Wanderung. §

Seite 2

Aufsatz

Schreiben Sie einen Aufsatz aus den folgenden Sätzen und Wörtern. Setze einen eigenen Titel.

während er ging, ging etwas kaputt

lächelt

Der Computer

55

Feargus der Ungestüme oder meine Flucht aus der grausamen Realität

Ich sitze hier und schreibe einen Test. Und werde geduzt. Der Testdurchführer scheint sehr nett zu sein, allerdings belächelt er meine Einwände lediglich. Zudem darf ich weder über ihn, noch über das Durchführungsobjekt, seinen heiligen Computer, schlecht reden. Deshalb wende ich mich nun von der harten Realität ab und flüchte in die Fantasiewelt.

Es war einmal ein Drache namens Feargus. (Es wäre nun wohl an der Zeit die Zahl 55 einzubringen. Aber was ist das denn für eine Zahl ?! Kein Drache ist 55 Jahre alt! Nun gut, ich darf mich ja nicht beschweren. Also weiter im Text.) Mit seinem zarten Alter von 55 Jahren (nicht Kartoffeln!), war seine Koordination noch nicht vollends ausgebildet, er war sehr schusselig. In der Drachenwelt wurde er deshalb auch Feargus der Ungestüme genannt. In seiner Familie, wie auch in seinem Freundeskreis machte man sich über ihn und seine Unfähigkeit seinen langen, eigentlich anmutigen Körper, zu bewegen lustig. Dies kränkte und betrübte ihn zugleich. So kam es eines schönen Abends zu einer Ausschreitung. Feargus sass mit seiner Familie gemütlich am Tisch, als ihm bei den, von lebhaften Gesten untermalten Ausführungen der Erlebnisse seines Tages sein Kelch hinunterfiel. Der ganze Tisch lachte. Doch er hatte genug. Entrüstet stürmte er aus dem Sahl. Während er ging, ging etwas kaputt, was nur noch von mehr Gelächter begleitet wurde.

Julia

Sonntag, 31. Juli 2016

Aufsatz

Schreibe einen Aufsatz aus den folgenden Sätzen und Wörtern. Setze einen eigenen Titel.

während er ging, ging etwas kaputt

lächelt

Der Computer

55

Oh, du schöne Zeit

Der Computer startete an diesem Tag von selbst. Er brauchte etwas Zeit um zu entscheiden, wo er beginnen sollte. Begrüssung an den Besitzer? Schönes Bild auf den Bildschirm? Dateien ordnen? Musik abspielen? Weckmelodie? Er überlegte und sinnierte und las ein wenig in der Zeitung....welch seltsame Zeit. Aber nach 55 Min begann die Arbeit.

Der Besitzer erwachte. Stress! Mail! Musik! Hausaufgaben vergessen? Was war nochmal zu tun!? Neue Grafikkarte funktioniert nicht! Hilfe!!!!!!

Der Besitzer hingegen lächelt. Er freut sich. Die Sonne scheint. Der Bach rauscht. Der Compi funktioniert. Herzli im Whatsapp. So ist der Morgen schön. Und jetzt? Wie geht denn das im Leben? Man geht zur Schule, tagaus, tagein....Die Welt dreht sich, tagaus, tagein.....die Katze miaut, tagaus, tagein.....Manchmal aber lächelt ein Mensch einfach.

Der Besitzer spaziert zum Zimmer hinaus, langsam. Die Katze streicht um seine Beine. Er dachte: Während er ging, ging etwas kaputt. Aber es ist nicht so. Es ist nur so, dass sich Menschen verändern. Nicht jeder Tag ist gleich. Jeder Tag ist anders. Nur ein wenig. Und jetzt? Man geht zur Schule – ein wenig anders....Die Welt dreht sich - ein wenig anders..., die Katze miaut.....ein wenig anders.....

So ist das!

Kyo Mangold

Dienstag, 12. Juli 2016

Aufsatz

Schreibe einen Aufsatz aus den folgenden Sätzen und Wörtern. Setze einen eigenen Titel.

während er ging, ging etwas kaputt

lächelt

Der Computer

55

Der alte Mann und sein Computer

Ein älterer Herr sitzt völlig erstaunt vor seinem Computer. Er ist begeistert von dieser neuen Technologie und wundert sich, warum diese Wundermaschine vor 55 Jahren noch nicht existierte. Diese Erfindung, der sogenannte Computer, hätte ihm Vieles erleichtert, er kann nun online einkaufen ohne das Haus zu verlassen, er kann sich über Alles und Jeden informieren innert Sekunden und er kann ständig mit seinen Kindern, Freunden und Verwandten kommunizieren. Es läuft jedoch nicht immer alles rund, § wenn er seinen Computer benutzt. Vor einigen Tagen ging seine Festplatte kaputt und er musste mit seinem Rechner zum Computerladen. Er machte sich folglich auf den Weg. Draussen regnete es ununterbrochen und er machte sich sorgen, ob er heil ankommen würde, denn er hat doch schon ein paar Jahrzehnte auf dem Buckel. Während er ging, ging etwas kaputt, er kümmerte sich nicht weiter darum und erreichte schliesslich den Computerladen. Kurz darauf wurde er von einem höflichen jungen Mann bedient und unterstützt. Dieser entdeckte dann, dass nicht nur die Festplatte, sondern auch das CD Laufwerk kaputt gegangen ist. Der Computer wurde repariert und der alte Mann war überglucklich, dass er nun endlich wieder seinen geliebten Computer benutzen konnte.

Simon Walo

Montag, 25. Juli 2016

Aufsatz

Schreibe einen Aufsatz aus den folgenden Sätzen und Wörtern. Setze einen eigenen Titel.

während er ging, ging etwas kaputt

lächelt

Der Computer

55

Neville und der kaputte Computer

Eine fast wahre Geschichte

Es war einmal vor langer Zeit ein kleiner Junge in einer grossen Stadt. Der Junge hieß Neville und war 15 Jahre alt. Neville war ein grosser PC-Fan. Er wusste alles über Computer: Wie man sie baut, wie man sie programmiert und natürlich wie man sie bedient. Doch eines Tages passierte etwas, womit Neville nicht gerechnet hat...

Es fing alles damit an, dass Neville, wie jeden Sonntag-Morgen, am Computer spielte. Es war jedoch kein besonderer Tag, nein, es war der 55. Geburtstag seines Vaters. Als Neville bemerkt hat, dass sein Vater aufgestanden ist, ist er ins Wohnzimmer und hat ihm alles Gute gewünscht. Sein Vater lächelte, noch etwas müde vom Aufstehen. Danach ging Neville wieder ins Zimmer. Schliesslich hatte er noch eine Menge Monster zu besiegen heute. Doch etwas war anders als zuvor: Der Computer schien nicht mehr zu funktionieren. Neville wurde stutzig: Hatte etwa seine Schwester etwas damit zu tun? Schliesslich lassen die beiden Geschwister normalerweise keine Gelegenheit aus, einander auf den Geist zu gehen. Und als Neville noch im Zimmer war, hat alles funktioniert. Erst während er ging, ging etwas kaputt. Doch Neville dachte nicht allzu lange darüber nach und ging stattdessen lieber nach draussen, um Sport zu machen. Deswegen wurde Neville Sportler und lebt heute gesund und glücklich ohne seinen Computer.

© Simon Walo

9.7 Farbauswahl der Testpersonen

Fiona

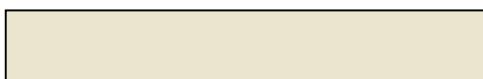
1.Farbe: 121,6,35



2.Farbe: 33,112,142



3.Farbe: 234,229,206



Kyo

1.Farbe: 0,0,128



2.Farbe: 0,191,255



3.Farbe: 33,33,33



Simon

1.Farbe: 0,255,195



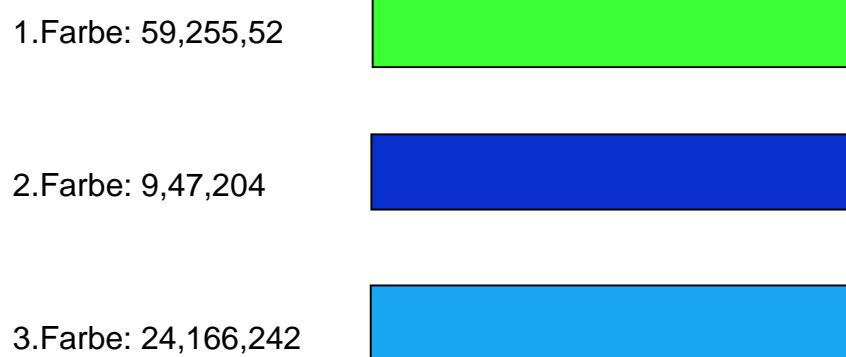
2.Farbe: 188,255,0



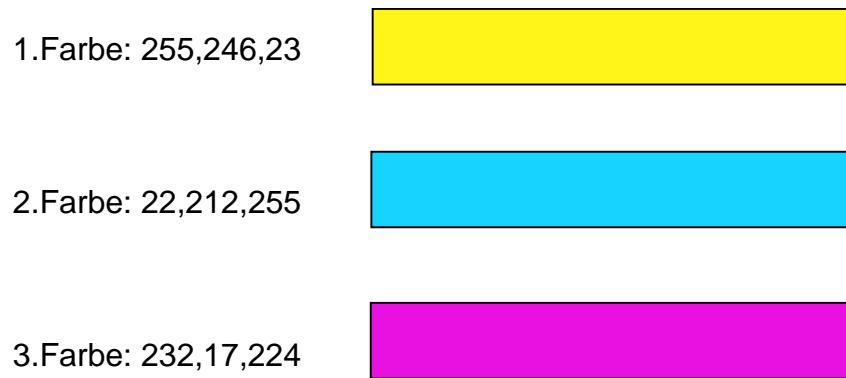
3.Farbe: 66,0,255



Andi



Julia



Gabriela

