UNIVERSITÄT
LEIPZIG

Softwaretechnikpraktikum

# Lastenheft


**Gruppe:** nw19a

**Mitglieder:** Thomas Pause, Sabine Lorius, Arik Korte, Martin George, Josephine Lange, Esther Prause, Anh Kiet Nguyen, Bärbel Hanle

**Verantwortlich:** Sabine Lorius & Josephine Lange

**Betreuer:** Dr. Nicolas Wieseke

**Tutor:** Martin Frühauf


**Abgabedatum:** 25.11.2019

**Stand:** 9. Dezember 2019

# Inhaltsverzeichnis

# 1 Visionen und Ziele

Ziel ist die Entwicklung einer Anwendung als Proof-Of-Concept, die es Poispielern ermöglicht zu berechnen, auf welche Weise beliebige Poifiguren erzeugt werden können (beschränkt auf den 2-dimensionalen Raum). Als einfachste Stufe sollen mit beliebigen Eingabeparametern sog. *Flowers* berechnet werden. Als weitere Funktionalität soll sowohl aus einer Touch-Eingabe als auch aus den Außenkonturen eines Eingabebildes eine parametrisierte Kurve erstellt werden können, die die gezeichnete Bahn eines Poikopfes darstellt. Auf dieser Grundlage soll eine neue parametrisierte Kurve errechnet und grafisch dargestellt werden, die die zur Erzeugung der Eingabekurve theoretisch nötige Handbewegung angibt. Den mathematischen Ansatz liefert hier die komplexe Fourieranalyse. Schließlich soll eine Simulation der Poibahn unter Eingabe der berechneten Handbewegungen und unter zusätzlicher Berücksichtigung physikalischer Aspekte erfolgen.

# 2 Rahmenbedingungen

Folgende Spezifikationen werden eingehalten:

- Codebasis: Java in Verbindung mit AndroidStudio
- Minimum API-Level: 25, Android 7.1.1 (Nougat)
- Kleinste unterstützte Display-Auflösung: HD (768 x 1366 Pixel)
- Lizenz: Open-Source (GPL/MIT)
- Dateiformat: APK

# 3 Kontext und Überblick

Der Auftraggeber möchte einen mathematisch-physikalisch fundierten Machbarkeitsbeweis für die Simulation von Poifiguren in Form einer Android-Anwendung. Es gibt bereits einige wenige Programme zur Simulation von Poifiguren, z.B. Poi Lab[1], VTG[2], VisualSpinner3D[3] und Poi Simulator[4], diese verfügen allerdings über keine Physik-Simulation und die Hand kann sich ausschließlich auf Kreisbahnen bewegen, welche einen fixen Radius im Verhältnis zur Poilänge haben. Gegenüber ähnlichen Anwendungen besteht eine innovative Funktionalität der zu entwickelnden Anwendung darin, aus einer beliebigen gegebenen Poibahn eine Handbewegung zu errechnen.

# 4 Funktionale Anforderungen

## 4.1 Muss-Ziele

**Bibliothek**

`/LFMB1/` Die App wird mit einer Reihe an vordefinierten Kurven, wie beispielsweise Herzen, Quadern o.Ä. ausgeliefert, die als Demonstration der Möglichkeiten der Anwendung dienen sollen. So bekommt der User direkt eine Vorstellung davon, was er mit der App machen kann.

**Flowers**

`/LFMF1/` Als Einstieg in den Bereich 'Flowers' gibt es einen entsprechenden Startbildschirm.

/LFMF2/ Zu Beginn soll eine mit voreingestellten Parametern initialisierte *Flower* präsentiert werden.

/LFMF3/ Nutzer können die Parameter zur Erzeugung von *Flowers* selbst definieren und ändern: $r$ (Radius), $\omega$ (Verhältnis der Frequenzen der Kreise), $\phi$ (Phasenverschiebung). Dafür ist eine manuelle Eingabe der Werte über die Tastatur zu ermöglichen.

/LFMF4/ Die durch Einstellung der Parameter definierte *Flower* kann graphisch dargestellt werden.

/LFMF5/ Auf Grundlage der Parameter ist zu berechnen, welche Handbewegung theoretisch nötig wäre, um eine bestimmte *Flower* zu erzeugen.

/LFMF6/ Es wird eine graphische Ansicht zur Verfügung gestellt, in der eine *Flower* und die komplette zugehörige Handbewegung gemeinsam zu sehen sind. Die Kurven von *Flower* und Hand sind dabei gut voneinander zu unterscheiden.

/LFMF7/ Nutzer sollen sich *Flower*form und Handbewegung im zeitlichen Verlauf als Animation abspielen lassen können, so dass zu jedem Zeitpunkt die relativen Positionen von Hand und Poikopf eindeutig bestimmt werden können.

**Touchpadeingabe**

/LFMT1/ Vom Startbildschirm führt ein Button zu einer Frei-Hand-Zeichenfläche, auf der mit den Fingern beliebige Kurven eingezeichnet werden können.

/LFMT2/ Ein Clear-Button bietet die Möglichkeit die Zeichnung von vorn zu beginnen.

/LFMT3/ Ein Finish-Button bietet die Möglichkeit die Eingabe zu beenden und sie an den *Curve-To-Hand*-Modus zu übergeben.

**Curve-To-Hand**

/LFMH1/ Im *Curve-To-Hand*-Modus wird eine *parametrisierte Kurve* – bspw. erstellt per Touchpadeingabe oder geladen aus der Bibliothek – als gewünschte Poibahn interpretiert, welche als Berechnungsgrundlage für die dafür erforderliche Handbewegung dient. Diese Handbewegung ist zu berechnen.

/LFMH2/ Für die kombinierte Darstellung von Poi- und Handbewegung sind vergleichbare Visualisierungsmöglichkeiten zur Verfügung zu stellen wie im Bereich 'Flowers' (graphische Ansicht, Animation).

## 4.2 Kann-Ziele

**Global**

/LFKG1/ Ein *Navigation Drawer* bietet jederzeit die Möglichkeit, in andere Bereiche der App zu wechseln.

/LFKG2/ Der *Navigation Drawer* zeigt andere Optionen an, je nachdem in welcher View der App er aufgerufen wird

/LFKG3/ Ein Share-Button bietet an geeigneten Stellen die Möglichkeit Graphiken/Videos/Berechnungen zu teilen.

/LFKG4/ Die Koordinaten einer generierten Kurve (als *Flower* oder über Touchpad-/Bildeingabe gewonnen) können in der Bibliothek abgespeichert werden.

/LFKG5/ Die graphische Darstellung einer generierten Kurve kann als Bilddatei in der Bibliothek abgespeichert werden (bei *Flowers* ggf. mit Angabe ihrer Parameterwerte auf dem Bild).

/LFKG6/ Eine graphische Darstellung, auf der sowohl eine als Poibahn zu interpretierende Kurve (gewünscht oder physikalisch berechnet) als auch die zugehörige Handbewegung zu sehen sind, kann als Bilddatei in der Bibliothek abgespeichert werden.

/LFKG7/ Die Animation einer kombinierten Poi- und Handbewegung kann als Videodatei in der Bibliothek abgespeichert werden.

/LFKG8/ Berechnungen und resultierende Grafiken/Animationen aus den Funktionen Bild-/Toucheingabe können in der Bibliothek gespeichert werden.

**Flowers**

/LFKF1/ Zusätzlich zur Eingabe der Parameterwerte über die Tastatur soll ein Schieberegler zur Veränderung der Werte eingebaut werden.

/LFKF2/ Während der Betätigung der Schieberegler für $r$ und $\phi$ kann der Nutzer live mitverfolgen, wie Parameteränderungen die Form der *Flower* beeinflussen.

/LFKF3/ Ein Geschwindigkeitsregler soll ermöglichen, die Geschwindigkeit der Animation zu erhöhen oder zu verringern.

/LFKF4/ Nutzer können die Animation pausieren und weiterlaufen lassen.

/LFKF5/ Von der 'Flowers'-Startseite aus soll eine gespeicherte *Flower* geladen werden können, um sie den Funktionalitäten des 'Flowers'-Bereichs zuzuführen (Parameteränderung und Neuberechnung der *Flower*form (LFMF3), kombinierte Ansichten von *Flower* und Handbewegung (LFMF6, LFMF7)).

/LFKF6/ Eine generierte *Flower* kann in der Bibliothek abgespeichert werden. In der einfachsten Form werden nur die Parameter abgespeichert.

**Direkteingabe einer parametrisierten Kurve**

/LFKD1/ Zur weiteren Bearbeitung im *Curve-To-Poi-* bzw. *Curve-To-Hand*-Modus kann eine *parametrisierte Kurve* (abgespeichert in einem passenden Dateiformat) aus der Bibliothek geladen werden.

**Touchpadeingabe**

/LFKT1/ Über einen Button kann die Frei-Hand-Zeichnung der Kurve durch eine Punkt-zu-Punkt-Eingabe ersetzt werden.

/LFKT2/ Mit erweiterten Optionen besteht die Möglichkeit hilfreiche Tools wie Koordinatensysteme als Overlay über der Zeichenfläche einzublenden.

/LFKT3/ Die Kurve kann via Drag & Drop punktweise verschoben und angepasst werden.

**Bildeingabe**

/LFKB1/ Mit der Systemkamera kann aus der App heraus ein Foto aufgenommen werden.

/LFKB2/ Per Zugriff auf die lokal gespeicherten Bilder des Users kann eine Kopie des Bildes zur Weiterverarbeitung importiert werden.

/LFKB3/ Das Bild wird intern so verarbeitet, dass Konturen abgebildeter Objekte stark hervorgehoben werden.

/LFKB4/ Bestimmte Parameter des *Image-Preprocessing*[5] sind beeinflussbar.

/LFKB5/ Der User hat die Möglichkeit aus den erkannten Konturen denjenigen Bereich zu wählen, der als Grundlage für die Eingabekurve herangezogen werden soll.

**Kombinierte Bild- und Touchpadeingabe**

/LFKK1/ Es soll möglich sein, bei der Touchpadeingabe ein Foto/Bild unterzulegen, welches dem Nutzer als Unterstützung beim Zeichnen der gewünschten Kurve dient. Mit dem Foto als Hintergrund kann der Nutzer auf dem Touchpad Ankerpunkte setzen und sich diese zu einer Kontur verbinden lassen.

/LFKK2/ Vor dem Laden in die Touchpadeingabe können Bereiche des ausgewählten Bildes ausgeschnitten und vergrößert werden.

**Curve-To-Hand**

/LFKH1/ Die im *Curve-To-Hand*-Modus errechnete Handbewegung kann als *parametrisierte Kurve* in die physikalische Simulation des *Curve-To-Poi*-Modus eingespeist werden, damit der User sich Abweichungen zur gewünschten Poibahn, welche durch die realistischeren Bedingungen möglicherweise auftreten, ansehen kann.

/LFKH2/ Für die Animation von Poi- und Handbewegung gelten dieselben Erweiterungsmöglichkeiten wie im 'Flowers'-Bereich (Geschwindigkeitsregler (LFKF3), Pausieren/Weiterspielen (LFKF4)).

**Curve-To-Poi**

/LFKP1/ Aus einer *parametrisierten Kurve*, die die Handbewegung darstellt, soll die Poibahn unter Berücksichtigung physikalischer Einflüsse errechnet werden.

/LFKP2/ Voreinstellungen für benötigte Parameter sollen es dem User erlauben, sich die Poisimulation anzeigen zu lassen, auch ohne sich mit der Bedeutung der Parameter auseinanderzusetzen.

/LFKP3/ Die Parameter für die physikalische Simulation sind einstellbar.

**Bibliothek**

/LFKA1/ (Vor-)gespeicherte Koordinaten, Graphiken und Animationen sind von hier jederzeit abrufbar.

/LFKA2/ Verschiedene Oberkategorien bieten einen schnellen Überblick.

/LFKA3/ Thumbnails geben in einer Art Galerie Vorschaubilder zu einzelnen *Flowers*/Kurven.

/LFKA4/ Dateinamen können bearbeitet werden.

**Einstellungen**

/LFKE1/ Über einen Schalter kann ein Dark-Mode de-/aktiviert werden.

/LFKE2/ Eine Möglichkeit zur Spracheneinstellung stellt dem Benutzer Deutsch und Englisch zur Wahl. Die Standardeinstellung ist Englisch, da dies die Hauptsprache in der Poigemeinschaft ist.

/LFKE3/ Ein Performance-Mode, der auf aufwändige Design-Elemente und Bildschirmübergänge verzichtet, soll die Anwendung auch auf älteren Geräten gut benutzbar machen.

**Hilfe**

/LFKS1/ Im Bereich 'Hilfe' werden Inhalte zugänglich aufbereitet um das Verständnis und die Bedienung der App zu erklären.

/LFKS2/ Ein kurzes Tutorial kann für die Erstbenutzung der App zur Verfügung gestellt werden.

# 5 Nichtfunktionale Anforderungen

**Bedienbarkeit/Benutzerfreundlichkeit**

`/LNB1/` Die sinnvolle Einbettung nativer Android Bordmittel (z.B. Kamera, Fotogalerie für den Bildimport) soll die intuitive Bedienung erleichtern.

`/LNB2/` Um auch ohne mathematische oder physikalische Kenntnisse ein zufriedenstellendes Ergebnis zu erhalten, soll der Nutzer mit Text und Bildern durch die Menüs geführt werden.

`/LNB3/` Der Bereich der Bildbearbeitung soll trotz detaillierter Einstellungsmöglichkeiten übersichtlich und verständlich bleiben, um den Nutzer nicht zu überfordern.

`/LNB4/` Nutzern soll eine kurze Erklärung, inwiefern *Flowers* durch die Parameter $r$, $\omega$ und $\phi$ definiert sind, zur Verfügung gestellt werden.

**Design**

`/LND1/` Ein einheitliches, gut durchdachtes Farbschema soll das Interesse des Nutzers steigern und das Anwendungserlebnis ansprechend gestalten.

`/LND2/` Animierte Bildschirmübergänge sollen die App modern wirken lassen.

`/LND3/` Für die Dauer von Lade- oder Berechnungszeiten soll dem Nutzer ein visuelles Feedback z.B. in Form einer Ladeanimation gegeben werden.

# 6 Qualitätsmatrix nach ISO 25010

| Kategorie | hoch | mittel | niedrig | nicht anwendbar |
|---|---|---|---|---|
| Funktionalität | | x | | |
| Zuverlässigkeit | | x | | |
| Effizienz | | x | | |
| Sicherheit | | | x | |
| Kompatibilität | x | | | |
| Benutzbarkeit | x | | | |
| Wartbarkeit | | x | | |
| Portierbarkeit | | | | x |

# 7 Lieferumfang und Abnahmekriterien

## 7.1 Lieferumfang

Die Anwendung wird als *APK-Datei*[6] ausgeliefert, um sie direkt für mobile Endgeräte verfügbar zu machen. Ein Benutzer- und Installationshandbuch, dokumentierter Quelltext sowie ein kurzer Testbericht sind ebenfalls im Lieferumfang enthalten.
Die Releasebündel werden als ZIP-Dateien gepackt und ausgeliefert.

### 7.2 Abnahmekriterien

Zum endgültigen Release wird die vollständige Umsetzung der unter Muss-Ziele zusammengefassten funktionalen Anforderungen garantiert. Insbesondere wird also die grundlegende Anforderung der Simulation von Poifiguren auf Grundlage komplexer Fouriertransformation in Form einer funktionsfähigen Android-Anwendung geliefert. Flowersimulationen und eine Zeichenfläche für die Kurveneingabe runden das Paket ab. Den Anforderungen der in der Qualitätsmatrix festgelegten Qualitätsanforderungen wird dabei stets Rechnung getragen.

## 8 Vorprojekt

Im Vorprojekt soll damit begonnen werden, den 'Flowers'-Bereich umzusetzen. Folgende Ziele müssen dabei erreicht werden:

- Aus den Parametern $r$, $\omega$ und $\phi$ kann eine *Flower*form berechnet werden, die dem Nutzer graphisch präsentiert wird (LFMF4).
- Der Nutzer kann die Parameterwerte selbst verändern und sich das Ergebnis ansehen (LFMF3 und LFMF4).
- Die zu einer *Flower* gehörige Handbewegung eines Poispielers wird ohne Berücksichtigung physikalischer Gesetzmäßigkeiten berechnet und zusammen mit der *Flower* graphisch präsentiert (LFMF5 und LFMF6).

Zusätzlich erstrebenswert, jedoch nicht zwingend erforderlich sind folgende Features:

- Erklärung zu den Parametern der *Flowers* (LNB4)
- Schieberegler zur Parameteränderung (LFKF1)
- Dynamischer Aufbau der *Flower* bei Parameteränderung (LFKF2)
- Animation mit Hand- und Poikopfbewegung (LFMF7)
- Geschwindigkeitsregler (LFKF3)
- Animation pausieren/weiterlaufen lassen (LFKF4)

# 9 Glossar

**APK-Datei** APK steht für Android Package. Es handelt sich hierbei um eine Installationsdatei für Android-Apps. Sie können entweder manuell oder beispielsweise über den Google Play Store automatisch installiert werden.

**Curve-To-Hand** Mit Curve-To-Hand wird ein Teilprogramm der App bezeichnet, das aus einer gegebenen *parametrisierten Kurve*, die die gewünschte Poibahn darstellt, mit Hilfe von *komplexer Fourieranalyse* eine andere parametrisierte Kurve errechnet, die angibt, welche Bewegung die Hand ausführen muss um diese Poibahn zu erzeugen. Dabei werden physikalische Aspekte außer Acht gelassen.

**Curve-To-Poi** Mit Curve-To-Poi wird ein Teilprogramm der App bezeichnet, das aus einer gegebenen *parametrisierten Kurve* (die hier eine Handbewegung darstellt) und aus den Anfangsbedingungen für die Position und Geschwindigkeit des Poikopfs auf Grundlage der *Newtonschen Bewegungsgleichung* eine andere parametrisierte Kurve errechnet, die angibt, auf welcher Bahn sich der Poikopf bewegt.

**Flower** Eine Flower ist eine spezielle *parametrisierte Kurve*, die durch eine Gleichung der Form

$$x(t) = cos(2\pi t) + r * cos(2\pi\omega t + \phi), \quad y(t) = sin(2\pi t) + r * sin(2\pi\omega t + \phi)$$

dargestellt werden kann. Dabei sind $r$, $\omega$, und $\varphi$ frei wählbare Parameter.
Ein Poispieler kann solch eine Figur erzeugen indem er die Hand, welche den Poi hält, in einer zusätzlichen Kreisbahn bewegt. Dabei entstehen in Abhängigkeit von Rotationsrichtung und Geschwindigkeit unterschiedliche Muster. Sie werden Flowers genannt, da viele dieser Figuren schlaufenförmige Blütenblätter, sog. 'Petals', aufweisen.

**Gaußsche Zahlenebene** Die Gaußsche Zahlenebene, auch komplexe Ebene, ist eine geometrische Repräsentation der komplexen Zahlen. Sie stellt jede komplexe Zahl als Punkt in dieser Ebene dar, wobei die $x$-Achse die Teilmenge der reellen Zahlen (also den Realteil) und die $y$-Achse die Teilmenge der rein imaginären Zahlen (also den Imaginärteil) repräsentiert. Eine komplexe Zahl $z = a + bi$ ist also der Punkt mit dem $(x, y)$-Tupel $(a, b)$.

**Image Preprocessing** Die digitale Bildvorverarbeitung dient dazu, eine bereits digitalisierte Bildvorlage zu bearbeiten, um darin enthaltene Informationen für den menschlichen Betrachter besser visuell sichtbar zu machen (Bildverbesserung) oder für die weiteren Verarbeitungsstufen der rechnergestützten Mustererkennung (Merkmalsextraktion und Klassifikation) aufzubereiten. Dies kann zum Beispiel durch Kantenschärfung, Kontrastverbesserung oder Helligkeitsanpassung geschehen.

**Komplexe Fourieranalyse** Komplexe Fourieranalyse ermöglicht es, eine *parametrisierte Kurve* in der *Gaußschen Zahlenebene* als eine Überlagerung von Kreisbewegungen mit ganzzahligen Frequenzen darzustellen.

**Navigation Drawer** Navigation Drawer bieten Zugriff auf Ziele und App-Funktionen. In der Regel werden sie über ein kleines Bildschirmsymbol aktiviert und als Overlay im aktuellen Menü eingeblendet.

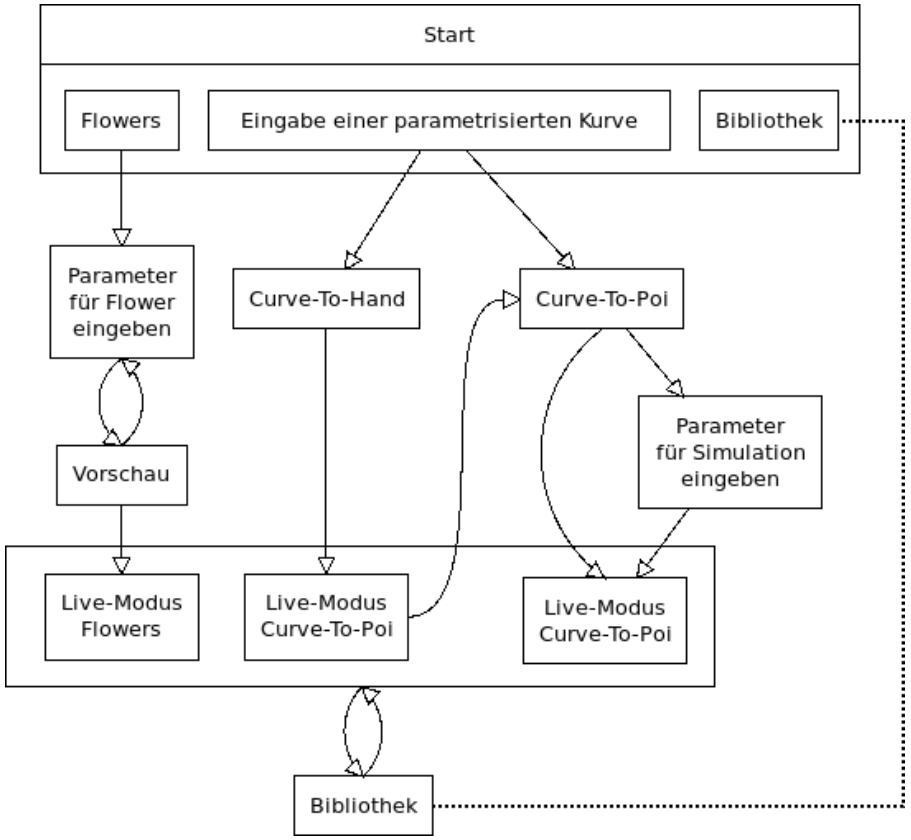
**Newtonsche Bewegungsgleichung** Die Newtonsche Bewegungsgleichung ist eine Differentialgleichung, die die Dynamik fester Körper beschreibt. Mit ihrer Hilfe lässt sich die Bahn eines Punktteilchens für alle Zeiten berechnen, wenn die Anfangsbedingungen (Ort und Geschwindigkeit) bekannt sind sowie die wirkenden Kräfte für jede Konfiguration berechnet werden können.

**Parametrisierte Kurve** Eine parametrisierte Kurve in $\mathbb{R}^n$ kann durch eine Funktion $\delta : M \subset \mathbb{R} \to \mathbb{R}^n$ dargestellt werden, wobei $M$ ein Intervall ist. Somit kann $t \to \delta(t)$ als Bewegung eines Punktes in $\mathbb{R}^n$ verstanden werden, wodurch von $\delta(M)$ eine durchlaufende Linie in $\mathbb{R}^n$ gezeichnet wird.

# A Strukturdiagramme

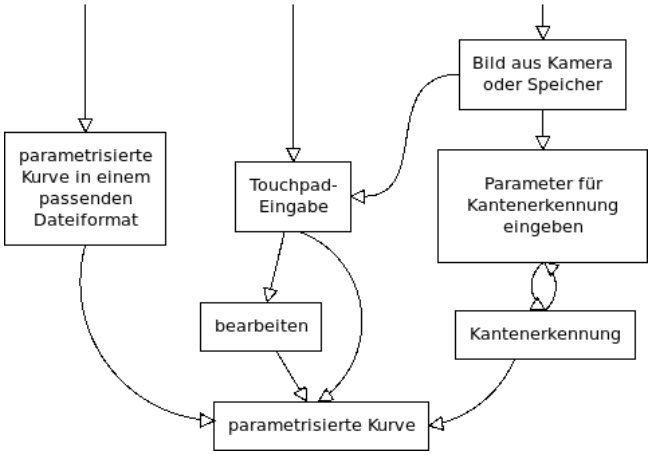
## A.1 Überblick

Das folgende Strukturdiagramm veranschaulicht die logische Wegeführung durch die App, dabei wird die Eingabe einer parametrisierten Kurve als Black Box (siehe A.2) betrachtet.

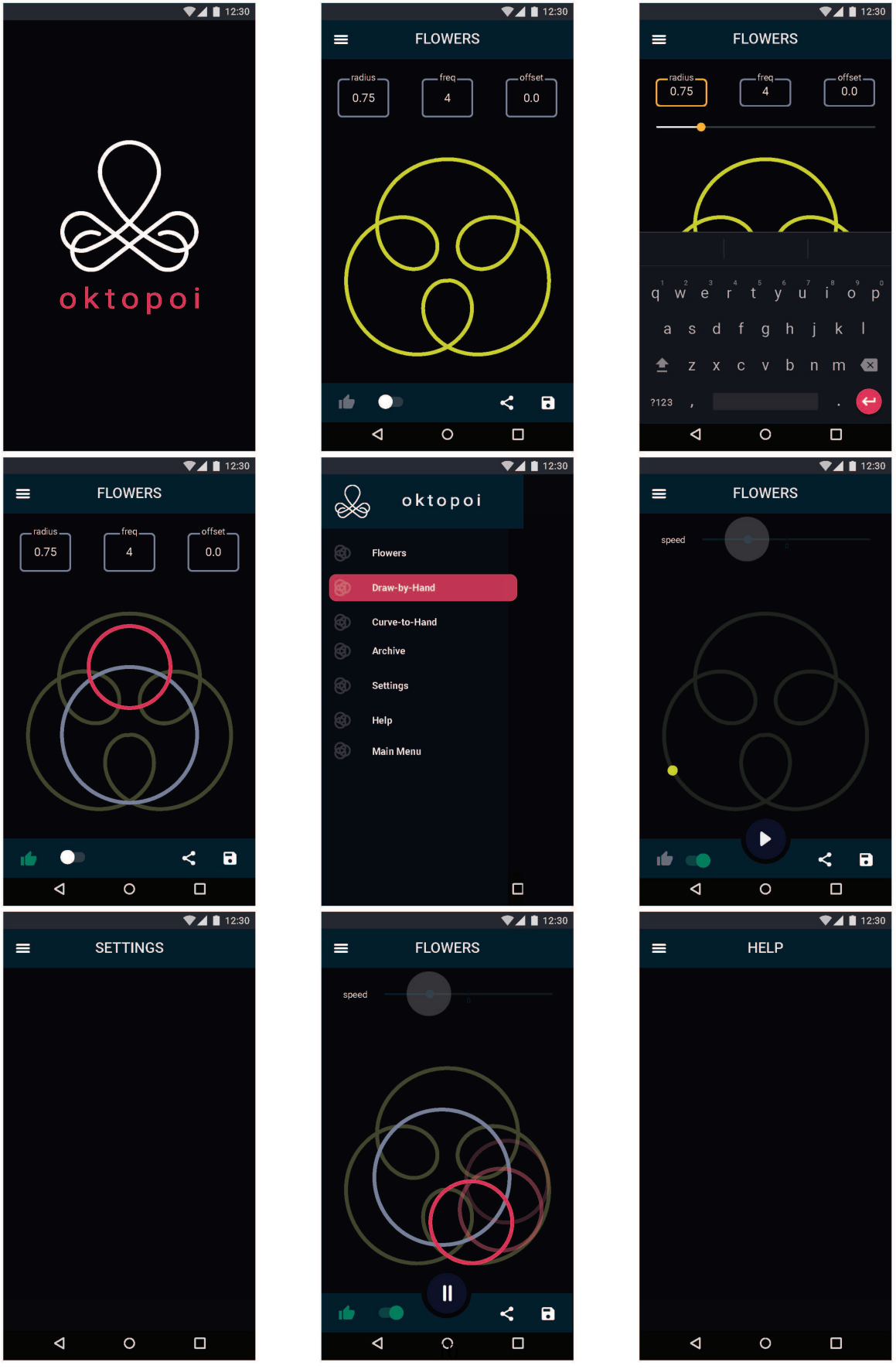



## A.2 Eingabe einer parametrisierten Kurve

Das nachstehenden Diagramm beleuchtet die unterschiedlichen Arten der Eingabe, die für das Erstellen einer parametrisierten Kurve herangezogen werden können.

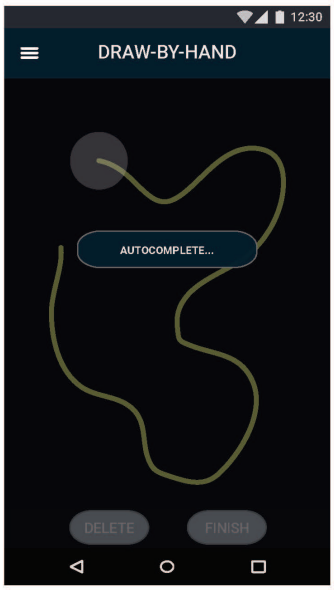


## A.3 Mockups


12:30
oktopoi
FLOWERS
radius
freq
offset
0.75
4
0.0
Flowers
Draw-by-Hand
Curve-to-Hand
Archive
Settings
Help
Main Menu
speed
SETTINGS
HELP

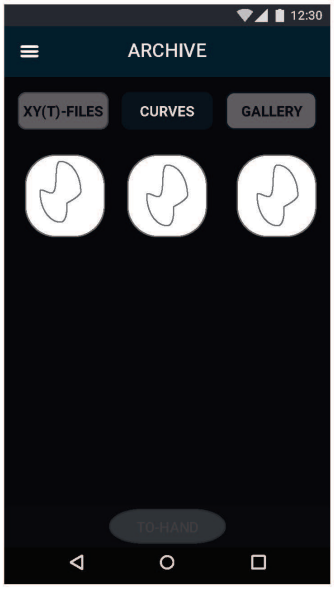

12:30
main menu
oktopoi
Flowers
Draw-by-Hand
Curve-to-Hand
Archive

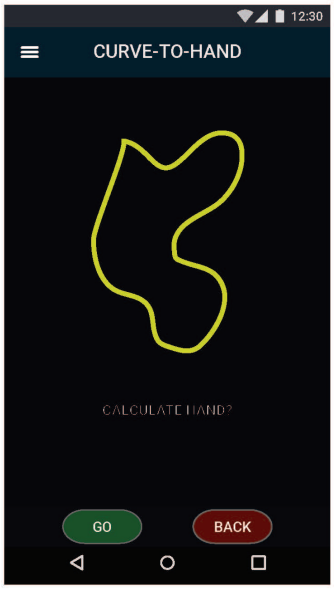

12:30
DRAW-BY-HAND
DELETE
FINISH


12:30
DRAW-BY-HAND
AUTOCOMPLETE...
DELETE
FINISH


12:30
DRAW-BY-HAND
TO-HAND


12:30
ARCHIVE
XY(T)-FILES
CURVES
GALLERY
FILE1.XY
FILE2.XY
FILE3.XY
FILE4.XY
FILE5.XY
FILE6.XY
FILE1.XYT
FILE2.XYT
FILE3.XYT
OPEN
TO-FLOWER


12:30
ARCHIVE
XY(T)-FILES
CURVES
GALLERY
TO-HAND


12:30
CURVE-TO-HAND
CALCULATE HAND?
GO
BACK


12:30
CURVE-TO-HAND
speed

# Quellenverzeichnis

[1] *Android-App zur Simulation von Poi-Figuren (Poi-Lab).* URL: https://play.google.com/store/apps/details?id=net.firestaff.mcp.poilab.free (besucht am 09. 11. 2019).

[2] *Android-App zur Visualisierung von Poi-Theorie (VRG).* URL: https://play.google.com/store/apps/details?id=mcp.firestaff.net.VTGv3 (besucht am 09. 11. 2019).

[3] *VisualSpinner3D.* URL: http://infiniteperplexity.github.io/visual-spinner-3d/demo.html (besucht am 22. 11. 2019).

[4] *Poi-Simulator von Danny Thomas.* URL: http://kaiein.com/poi_sim/sim6.htm (besucht am 09. 11. 2019).

[5] *Bildverarbeitungsprozess.* URL: https://de.wikipedia.org/wiki/Schwellenwertverfahren (besucht am 22. 11. 2019).

[6] *Was ist eine APK-Datei.* URL: https://praxistipps.chip.de/was-ist-ein-apk-einfach-erklaert_42098 (besucht am 20. 11. 2019).