



# Businessplan

für die Fertigstellung und Vermarktung der

## Grafiksoftware „Fantasijana“

### 0. Executive Summary

Es geht um die Entwicklung und Vermarktung eines Grafikprogramms. Angesichts der Vielzahl – zum Teil sogar kostenloser – Programme in diesem Markt erscheint dies zunächst als wirtschaftlich völliger Unsinn. Eine nähere Analyse ergibt jedoch, dass die Konkurrenz zwar zahlreich, aber grottenschlecht ist. Dies betrifft teils die bereitgestellten Funktionen, teils auch die Bedienung der Programme – meist leider beides. Dazu kommt, dass entgegen der offenbaren Marktabdeckung eine genauere Betrachtung zeigt, dass für wesentliche Teilmärkte auch heute noch keine adäquate Software zur Verfügung steht.

*Im Gegensatz zur Konkurrenz wird es „Fantasijana“*

- *auch dem Anwender, der **keine jahrelange Ausbildung** als Grafik-Designer, Fotograf o.ä. hat,*
- *durch eine **unbegrenzte** Anzahl von*
- ***selbst-definierbaren** und*
- ***wieder-verwendbaren** Formen, Farben, Verläufen, Anordnungen etc. –*
- *bei einer **trotzdem überschaubaren** Programmoberfläche – ermöglichen,*

- **für jeden nur denkbaren Zweck** (*Kunst ebenso wie Wissenschaft, Lehre, Technik, Datenanalyse usw.*)
- **bildliche Darstellungen nach seinen Vorstellungen und Ideen**

zu gestalten.

Da zudem auf einen Prototyp zurückgegriffen werden kann, ist der (Rest-) Aufwand für die Entwicklung – gemessen an durchschnittlichen Softwareprojekten – geradezu hervorragend kalkulierbar und ausgesprochen niedrig. Durch das Ansprechen einer neuen und auch langfristig lukrativen Zielgruppe (Schüler etc.) soll Umsatz auf Dauer in einer Höhe generiert werden, die einen erheblichen wirtschaftlichen Erfolg auch in Zukunft erwarten lässt.

Anmerkungen:

1. Die Zielgruppe, die hier genannt wurde, ist nur die, die als erstes ins Auge gefasst werden soll. Mittel- bis langfristig sollen Märkte aller Art bedient werden, sofern dort irgendwie ein Bedarf an Visualisierung besteht – und das sind fast unbegrenzt viele.
2. Durch neuartige, spezielle Programmierkonzepte sollen Entwicklungs-, Test- und Wartungsaufwand klein gehalten werden. Durch die Integration von Anwender- und Programmiererdokumentation sollen aus Sicht der Entwickler einerseits stundenlanges Herumsuchen in Dokumentationen und andererseits Mehrfachaufwände und fehleranfällige Redundanzen eliminiert werden. Die weitgehende Standardisierung sowohl der Oberfläche, als auch des Programms sollen Mehrfachaufwände bei der Einarbeitung sowohl für die Benutzer, als auch für die Entwickler klein halten
3. Am Rande seien hier noch die Ideen hinter dem Namen kurz erläutert:
  - „Fantasie“ soll eine große Rolle spielen.
  - Außerdem enthält der Name natürlich eine Anspielung auf die Programmierung in Java. („jana“ statt „java“ soll so ein bisschen nach Frauennamen klingen – Muse, gute Fee oder so)
  - Abkürzen können Jugendliche selbst ganz gut. (z.B. „Fanta“: Die dürfen das – ich nicht!)
  - Das „i“ vor dem „j“ wurde eingefügt vorwiegend wegen möglicher Kunden im Spanisch sprechenden Raum. Dort wird das „j“ ähnlich wie im Deutschen „ch“ in „ach“ ausgesprochen, was hinter einem „s“ zum Zungenbrecher würde. Ähnliches gilt für andere Sprachen.

## Inhalt

<b>1. Der Gründer / Entwickler</b>	<b>3</b>
<b>2. Die Geschäftsidee: Grafiksoftware „Fantasijana“</b>	<b>7</b>

<b>3. Kunden</b>	<b>15</b>
<b>4. Markt und Wettbewerber</b>	<b>15</b>
<b>5. Strategie</b>	<b>16</b>
<b>6. Marketing / Werbung</b>	<b>16</b>
<b>7. Unternehmen</b>	<b>17</b>
<b>8. Finanzen</b>	<b>17</b>
<b>9. SWOT-Analyse</b>	<b>19</b>
<b>10. Meilensteine</b>	<b>19</b>
<b>11. Anhang: Beispielbilder, erzeugt mit dem Prototypen</b>	<b>20</b>

## **1. Der Gründer / Entwickler**

### **a. Lebenslauf**

#### **Persönliche Daten**

Geburtsdatum	27.12.1960
Geburtsort	Kamen
Mutter	Irene Kellerwessel †, geb. Wilski; erst Hausfrau, später Richterin am HR
Vater	Dr.-Ing. Hans Kellerwessel †, bis 1988 Abteilungsleiter F&E bei KHD
Familienstand	Ledig, keine Kinder

#### **Schulbildung**

1966 bis 1970	Grundschule in Köln-Porz
1970 bis 1971	Stadtgymnasium Porz
1971 bis 1979	Couven-Gymnasium Aachen

#### **Studium und Zivildienst**

06.1979 bis 08.1979	Praktikum für Maschinenbau bei der Klöckner-Humboldt-Deutz AG, Köln
10.1979 bis 02.1980	Maschinenbau an der RWTH Aachen (ohne Abschluss)
03.1980 bis 06.1981	Zivildienst in Stuttgart
10.1981 bis 03.1984	Betriebswirtschaft an der RWTH Aachen
04.1984 bis 11.1987	Betriebswirtschaft an der Universität Köln Schwerpunkte: Organisation, Unternehmensrechnung, Informatik Diplom mit Durchschnittsnote 2,6
10.1987 bis 09.1988	Physik und Mathematik an der Universität Köln (ohne Abschluss)
10.1988 bis 09.1989	Mathematik an der RWTH Aachen (ohne Abschluss)

03.1993 bis 01.1997 Operations-Research-Zusatzstudium an der RWTH Aachen  
Magister in Operations Research (M.O.R.) mit Durchschnittsnote „gut“;  
[Magisterarbeit mit dem Thema „Denk- und wahrnehmungs-  
psychologische Probleme beim computerunterstützten Entscheiden“](#)

### **Berufspraxis und Weiteres**

10.1986 bis 12.1992 Zunächst Praktikant, dann freier Mitarbeiter im Bereich  
Softwareentwicklung bei der Organisationsabteilung der Philips  
Kommunikations Industrie AG, Köln

09.1993 bis 07.1996  
und 03.1997 bis 05.1997 Projektleiter bei der WORKS EDV-Service GmbH, Aachen

06.1997 bis 05.1998 Analyst/Programmierer bei der Santander Direkt Bank AG, Frankfurt a.M.

08.1998 bis 03.1999 Softwareentwickler bei der SBM GmbH, Würselen

04.1999 bis 07.1999 Projektingenieur bei der Topsytem Systemhaus GmbH, Würselen

08.1999 bis 02.2004 Selbständig tätig als Softwareentwickler und EDV-Berater

seit 02.2004 Arbeitsunfähig durch Schlaganfall, seitdem durch rechtsseitige, fast  
vollständige Lähmung schwerbehindert, kann seit etwa 2010 jedoch  
wieder ohne Gehhilfen gehen. Grad der Schwerbehinderung: 70%.  
Beruflich fit gehalten durch Entwicklung von Tools mit Hilfe von Java.  
Seit etwa 2010 Entwicklung des genannten Grafikprogramms unter  
Ausnutzung von Java2D (Prototyp vorführbereit).

Sprachkenntnisse Englisch fließend, Grundkenntnisse in Französisch, Italienisch und  
Spanisch

Veröffentlichungen [Programmierrichtlinien in der Praxis](#) (Mitp-Verlag 2002)  
Übersetzung eines Buches über UML (Mitp-Verlag 2003)  
[Dape - Was Sie schon immer über die Softwarebranche geahnt  
haben, aber eigentlich lieber nicht wissen wollten \(eBook, ePUB\)](#)

Kenntnisse in den Bereichen Unternehmensrecht, Grafik, technisches und Informatikwissen u. a.  
habe ich zum großen Teil aus meinem sozialen Umfeld:

- meine Mutter hat als Richterin – unter anderem zuständig in Bereichen wie Handelsregister  
und Grundbuch – gearbeitet,

- mein Vater war Dr. Ing. (Bergbau) und Abteilungsleiter im F&E-Bereich eines international tätigen Maschinen- und Anlagenbaukonzerns; er arbeitete zeitweilig auch als Redakteur beim VDI-Verlag
- mein ältester Bruder hat Jura studiert und ein Traineeprogramm bei einer Bank gemacht,
- mein zweitältester Bruder ist Diplom-Graphik-Designer und Ausbilder Preprint (Druckvorstufe) beim Verband Druck und Medien in Oldenburg

Dazu kommen Berufskollegen aus der EDV / IT, Freunde aus Bereichen wie Physik und Kristall-Chemie (Farbstoffe sind meist kristallin), Photo- und Vermessungsingenieure usw.

## **b. Qualifikation**

- ➔ (Betriebs-) wirtschaft studiert; leicht überdurchschnittliches Diplom
- ➔ Aufbaustudium mit Schwerpunkten in den Bereichen Optimierung und Entscheidung; dabei auch intensive Beschäftigung mit Denk-, Wahrnehmungs- und Entscheidungspsychologie
- ➔ jahrelange Tätigkeit in fast allen Bereichen der Softwareentwicklung (Architektur, Datenbank- und Benutzungsoberflächendesign, Programmierung, Schreiben von Handbüchern – Ausnahme: Testen)
- ➔ viele Kenntnisse, die für die Entwicklung eines Grafikprogramms am Rande von Interesse sind.

## **c. Motivation**

Ein Bild sagt mehr als tausend Worte: Ich möchte jungen Menschen das Lernen – und damit einen wichtigen Teil ihres Lebens – leichter machen. Zudem möchte ich ihnen Spaß und Spiel bieten, ohne sie dabei zu einer Konsumentenhaltung zu verleiten.

Dazu entspringt meine Motivation auch der Tatsache, dass hier die Hürden für eine Softwareentwicklung ungewöhnlich niedrig sind – und damit die Erfolgsaussichten genauso ungewöhnlich hoch. (Eine Software selbst zu entwickeln und zu vermarkten wäre mir sonst viel zu riskant!)

- Ein Prototyp existiert bereits (grob geschätzt 15.000 LOC [nicht-leere Programmzeilen] in der Programmiersprache Java), die Machbarkeit des Ganzen ist gesichert, jahrelange Entwicklungsarbeit konnte ohne irgendwelchen Druck (also qualitätsorientiert!) bereits geleistet werden. Alle Abbildungen in diesem Businessplan wurden mit Hilfe dieses Prototypen erstellt (mit Ausnahme natürlich der Darstellung von Oberflächen anderer Programme).

- Konkurrenten gibt es viele, auch kostenlose – aber die haben ausnahmslos irgendwelche gravierenden Schwächen in Bereichen wie Bedienung, Funktionalität usw. (wahrscheinlich auch bei Interna, wie beispielsweise Lesbarkeit und Wartbarkeit des Quellcodes, Speicherplatzbedarf und dgl.).

Außerdem sind diese Programme bestenfalls zweckorientiert gestaltet, niemals auf eine Zielgruppe ausgerichtet. Die Empfehlung eines Freundes für gleich zwei Grafikprogramme mit völlig unterschiedlicher Benutzungsoberflächen zeigt drastisch auf, wie die Folgen vor allem für jene Anwender aussehen, die nicht schon jahrelang mit den Merkwürdigkeiten der Bedienung eigentlich gewohnter Programme vertraut sind.

Bei den erhältlichen Grafikprogrammen werden meist einfach blind irgendwelche Funktionen zusammengeschustert, die Programme oft überhaupt nicht strukturiert, überladen mit noch mehr Funktionen, die dann aber kein Mensch mehr in den geradezu utopisch fehl-strukturierten (gemessen an Erkenntnissen aus der Wahrnehmungspsychologie) Menüs mehr findet usw.

- Ich möchte eine neue Zielgruppe ansprechen: Schüler, Azubis und Studenten sollen zum einen ihrer Fantasie und ihrem Spieltrieb freien Lauf lassen können („homo ludens“) und zum anderen bei ihrer „Arbeit“ unterstützt werden (Anfertigung von Abbildungen für Studienarbeiten, verständliche Darstellung von Zusammenhängen usw.).

Wenn sie älter sind, sollen Zusätze sie in ihrem jeweiligen Metier unterstützen.

Natürlich sollen auch ihre jeweiligen Ausbilder (Lehrer an allgemeinbildenden und Berufsschulen, Professoren usw.) Unterstützung bekommen.

- Die Bedürfnisse der von mir angepeilten (Haupt-) Zielgruppe sind zumindest im ersten Angang besonders leicht und kostengünstig zu ermitteln: Beispiele für (gute) Lehrbücher gibt es zuhauf (notfalls US-amerikanische statt deutschen), wie man es richtig machen sollte kann man heutzutage überall im Internet nachlesen usw.

Und das, was die Fantasie braucht, weiß ich selbst oder werde es im Laufe der Zeit im Gespräch mit Anwendern ermitteln.

... und last but not least möchte ich natürlich nicht nur Dinge im betriebswirtschaftlichen Bereich und bei der Programmkonzeption besser als üblich machen, sondern will vor allem wissen, ob man nicht Geld mit etwas Sinnvollem verdienen kann, auch wenn man nur Software entwickeln gelernt hat.

## **2. Die Geschäftsidee: Grafiksoftware „Fantasijana“**

Die Grafiksoftware „Fantasijana“ – so der Projektname – ist nicht gedacht für Leute, für die sich eine – u.U. sogar aufwändige – Einarbeitung in ein Grafikprogramm lohnt. Sie soll nicht von Profis – wie Graphikern oder Photographen – eingesetzt werden, sondern von gelegentlichen, möglicherweise sogar im Umgang mit Windowssystemen wenig geübten Benutzern: in der Hauptsache Schülern und Studenten, Lehrern und Ausbildern, Auszubildenden oder sonstigen Personen, die eine -ruck-zuck erstellte Skizze benötigen, um jemandem einen Zusammenhang zu vermitteln o.ä.

Darüber hinausgehend ist es Ziel, das Arbeiten mit diesem Programm so spielerisch leicht zu machen, dass das Erstellen von Bildern hiermit sogar eine Spaß bringende und die Kreativität fördernde Freizeitbeschäftigung darstellt.

### **Kernpunkte**

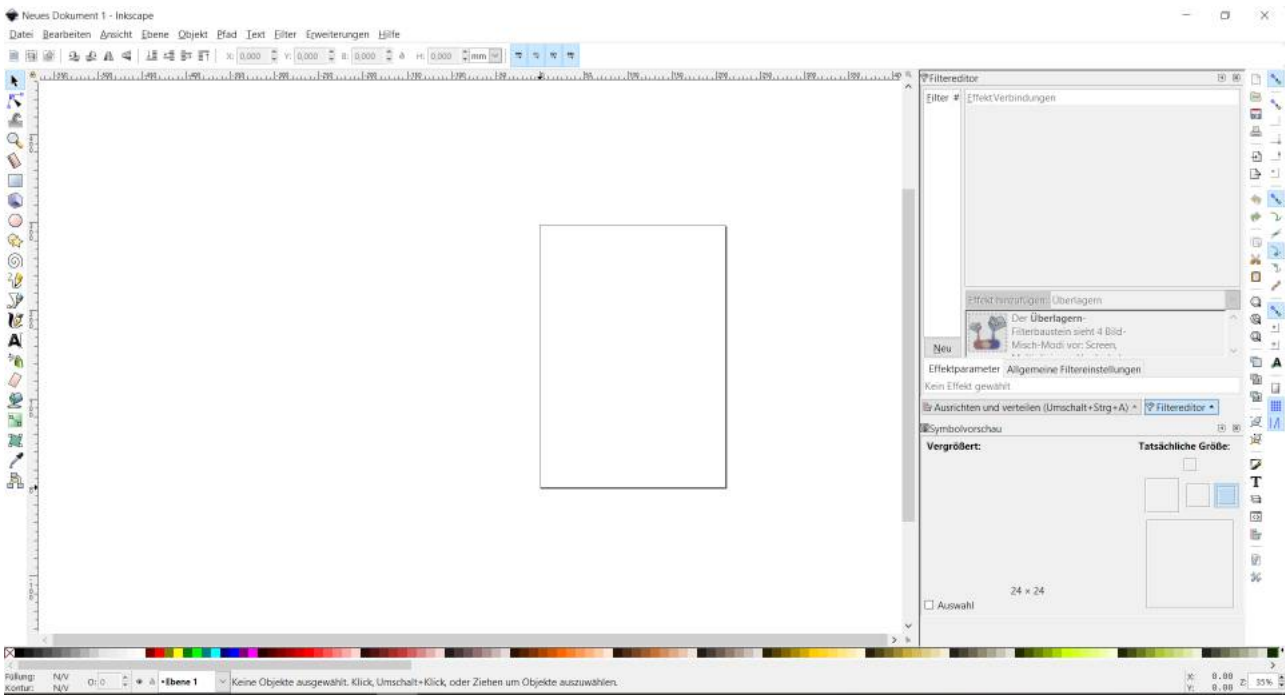
Die **Oberfläche** des Programms soll folgenden Regeln folgen:

1. Keine versteckten Funktionen
2. Jeder Funktion entspricht eindeutig nur ein Button. (Nicht Menüs und Symbolleisten und Tastenkombinationen und und und – der Funktionsaufruf geht bei Eindeutigkeit nämlich auf die Dauer schneller, weil er eingeübt wird.)
3. Keine Anpassung durch den einzelnen Benutzer möglich (wenn ich zum Schreibtisch des Kollegen gehe, möchte ich mich nicht in dasselbe Programm neu einarbeiten müssen!)
4. Soweit wie möglich (verständliche!!!) Symbole (Das Programm soll auch für jene nutzbar sein, die Deutsch noch nicht so beherrschen, Schüler/Studenten oder Fachkräfte aus dem Ausland oder so. Gedacht ist, statt – aufwändiger – Internationalisierung dem Programm eine Oberfläche in Deutsch, Englisch und „Zeichensprache“ zu geben.)
5. Bedienung nur mit Klicken auf eine Maustaste; kein Ziehen mit herunter gedrückter Taste; kein Klicken mit der Maus in Kombination mit Tasten der Tastatur (Strg, Ctrl, Alt, Shift, ...)
6. Mehrdeutige Klicks führen zu Rückfrage (Auswahl: „Möchtest du dies, das oder jenes?“).
7. Möglichst vollständiges Vermeiden von Menüs wie „Optionen“, „Extra“ und dergleichen
8. Keine Überforderung der Wahrnehmungsmöglichkeiten des Anwenders: Maximal sieben Unterpunkte darf es zu einem Punkt geben! (Äh ... das war der achte Punkt.)

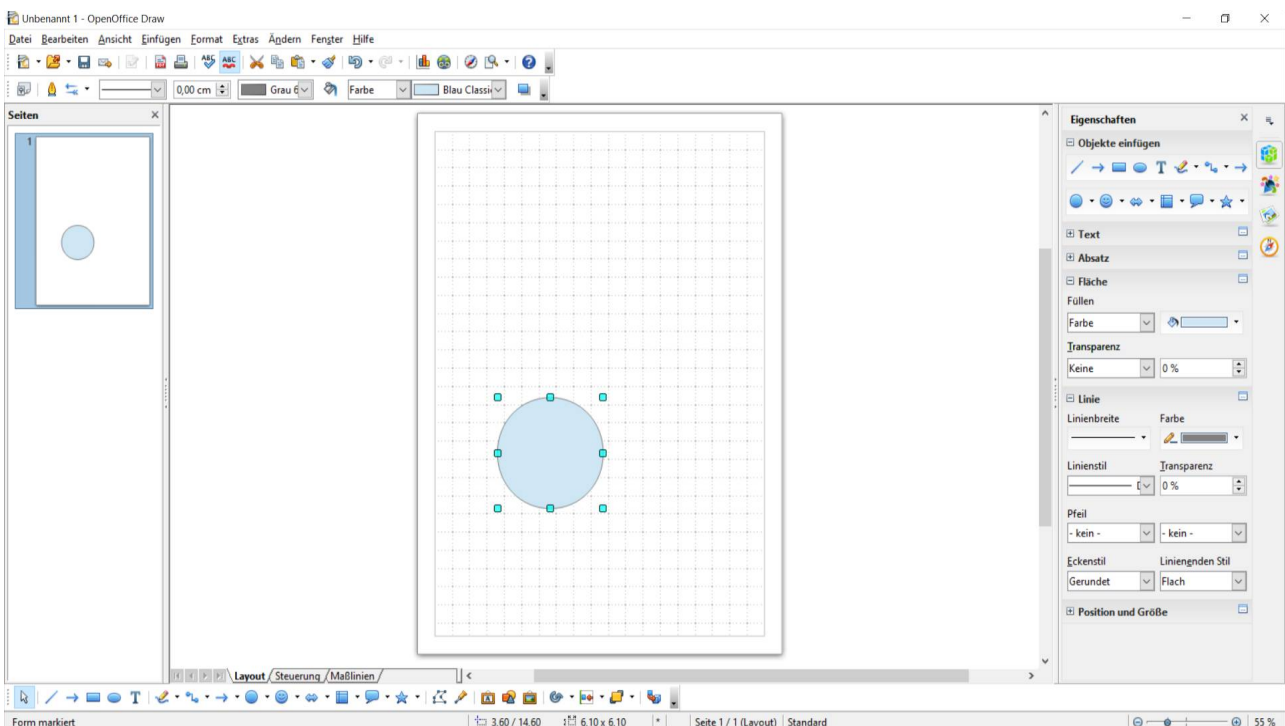
Realisiert werden soll dies durch vollständigen Verzicht auf ein Menü. Es gibt nur noch (möglichst nur sieben) Symbolleisten – also für jeden verständlich und nachvollziehbar – geordnet. Motto: Spaß statt ewigem Rumsuchen in Menüs! Das Programm ist für den Menschen gemacht und nicht umgekehrt!

Auf ein Handbuch oder eine generelle Hilfefunktion wird verzichtet: Bei jedem Anklicken eines Buttons kann man die aufgerufene Funktion auch wieder abbrechen; außerdem gibt es natürlich eine Hilfe zur konkret aufgerufenen Funktion. (Handbuch lesen tut 'eh keiner. Dafür ist auch keine Zeit im Tagesgeschäft oder während des Erstellens einer Zeichnung für eine Studienarbeit! Beim Einsatz im Unternehmen ist Einarbeitung ohnehin nur ein erheblicher Kostenfaktor.)

Beispiel 1 für eine völlig überfrachtete Oberfläche, Aufrufen ein- und derselben Funktion über Speedbuttons und Menüpunkte usw.:

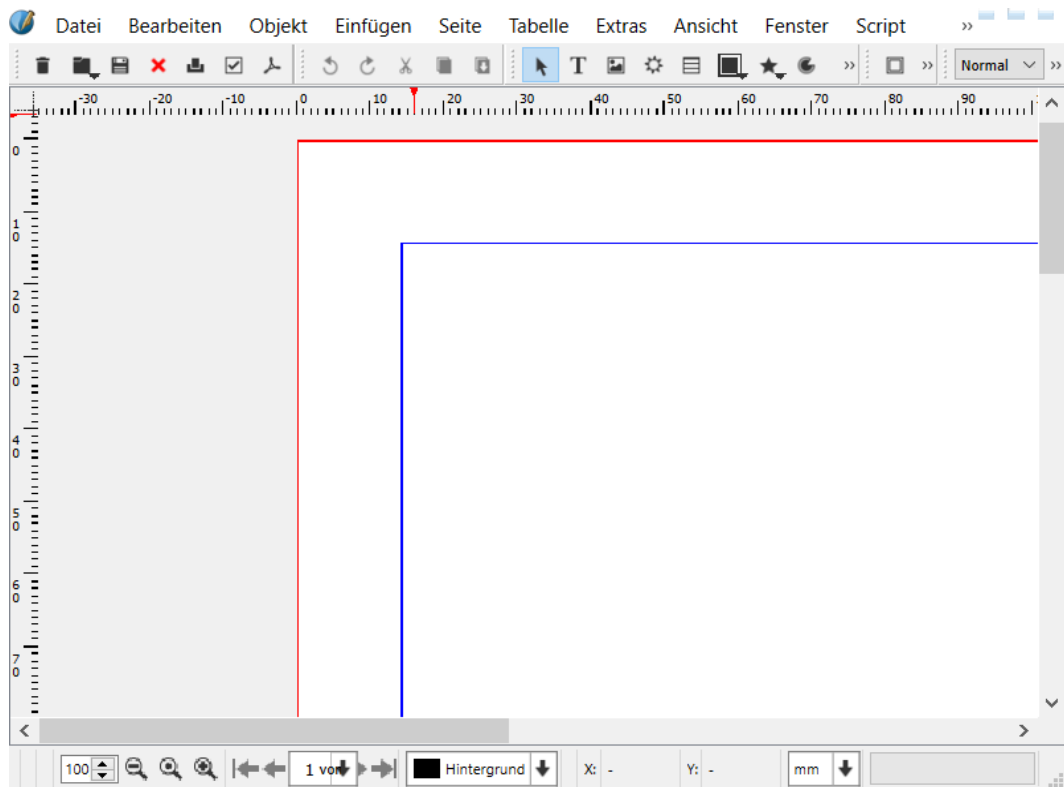


Beispiel 2 für eine völlig überfrachtete Oberfläche:





Beispiel 3: Nicht ganz so schlimm überfrachtet, allerdings enthält die – notdürftig unterteilte – Symbolleiste voll ausgefahren (also wenn das Fenster hinreichend breit gemacht wird) über 40 Punkte:



## **Funktionalität von Fantasijana**

Fantasijana soll mit einer Reihe von Alleinstellungsmerkmalen (USP's) aufwarten. Dabei soll darauf geachtet werden, dass es sich dabei jeweils nicht um „überflüssigen Schnick-Schnack“, sondern um Funktionen handelt, die den Anwendern der angepeilten Zielgruppe auch bei ihrer Arbeit nützen.

Übersicht:

### **flexiblere Layout-Funktionen**

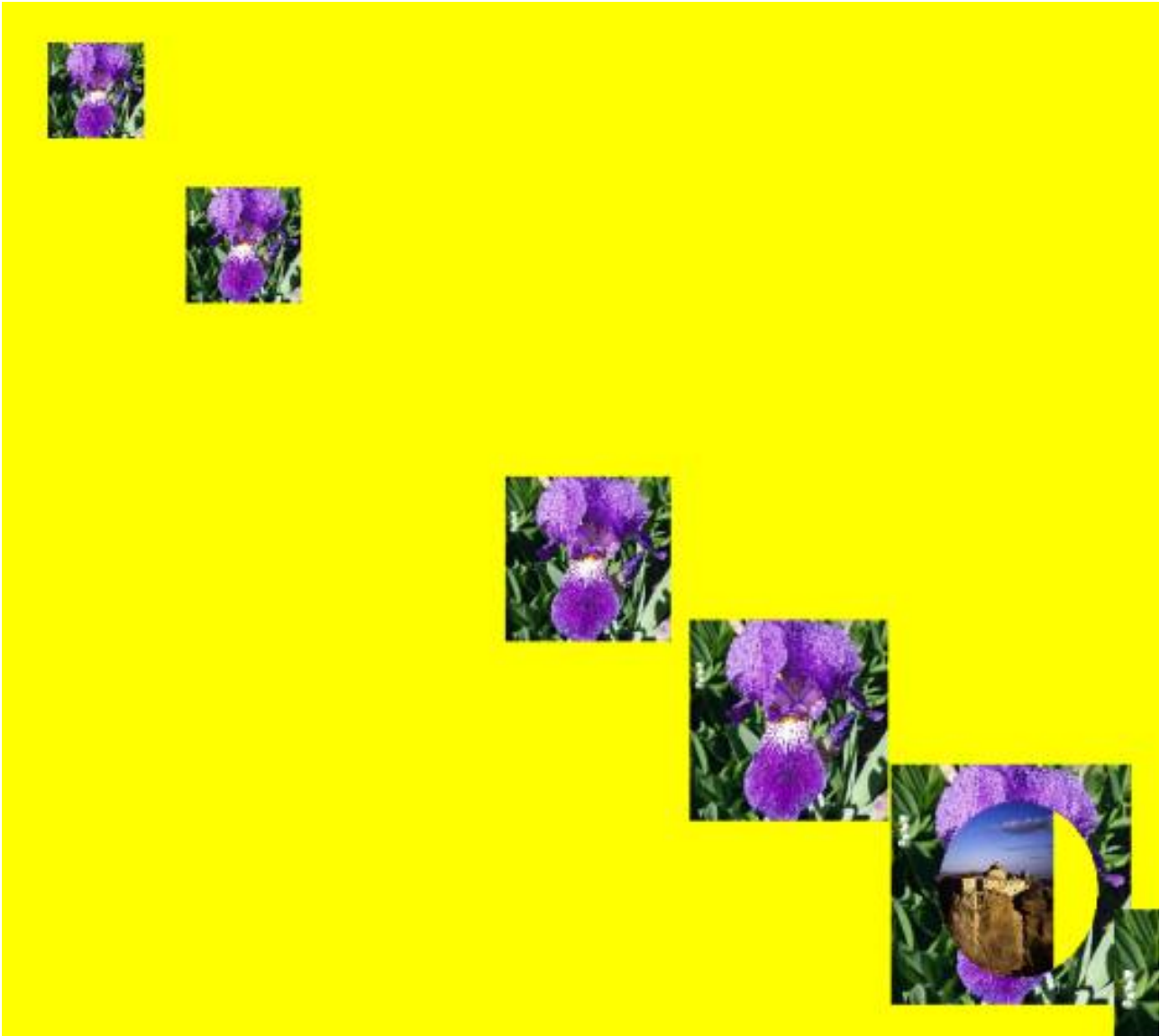
Üblich bei objektorientierten 2D-Grafikprogrammen ist, dass die sogenannte z-Order verändert werden kann (nach vorne – nach hinten) und dass die Objekte links, rechts, oben, unten oder zentriert auf einer Linie unter- oder nebeneinander angeordnet werden können. Was bisher m.W. kein Grafikprogramm kann – obwohl es sehr einfach ist – ist eine Anordnung in Kreis-, Spiral- oder Wellenform oder dergleichen. Dies soll „Fantasijana“ als erste Software am Markt ermöglichen. Weiterhin wird „Fantasijana“ die Manipulation von Layouts unterstützen („Affine Transformationen“, wie es der Mathematiker nennt: Drehen, Spiegeln, Scheren – bestimmte Art der Verzerrung, Vergrößern, Verkleinern, ...)

„Fantasijana“ soll es ermöglichen, anpassbare und **selbst definierbare „Silhouetten“** (selbständig, nicht als Teil eines Bildes!) zu speichern. Geplant ist dabei die (vollständige) Unterstützung von „CAG“ („Constructive area geometry“: Durchschnitt und Vereinigung von Flächen usw.). Auch Farben, Verläufe, sogenannte „wallpapers“ (gekachelte Bilder, wie sie z.B. als Desktophintergründe benutzt werden), Strichelungen für Umrisse usw. sollen vom Benutzer definiert und gespeichert werden können. Wiederverwendbarkeit aller selbstdefinierten Objekte ist das Ziel!

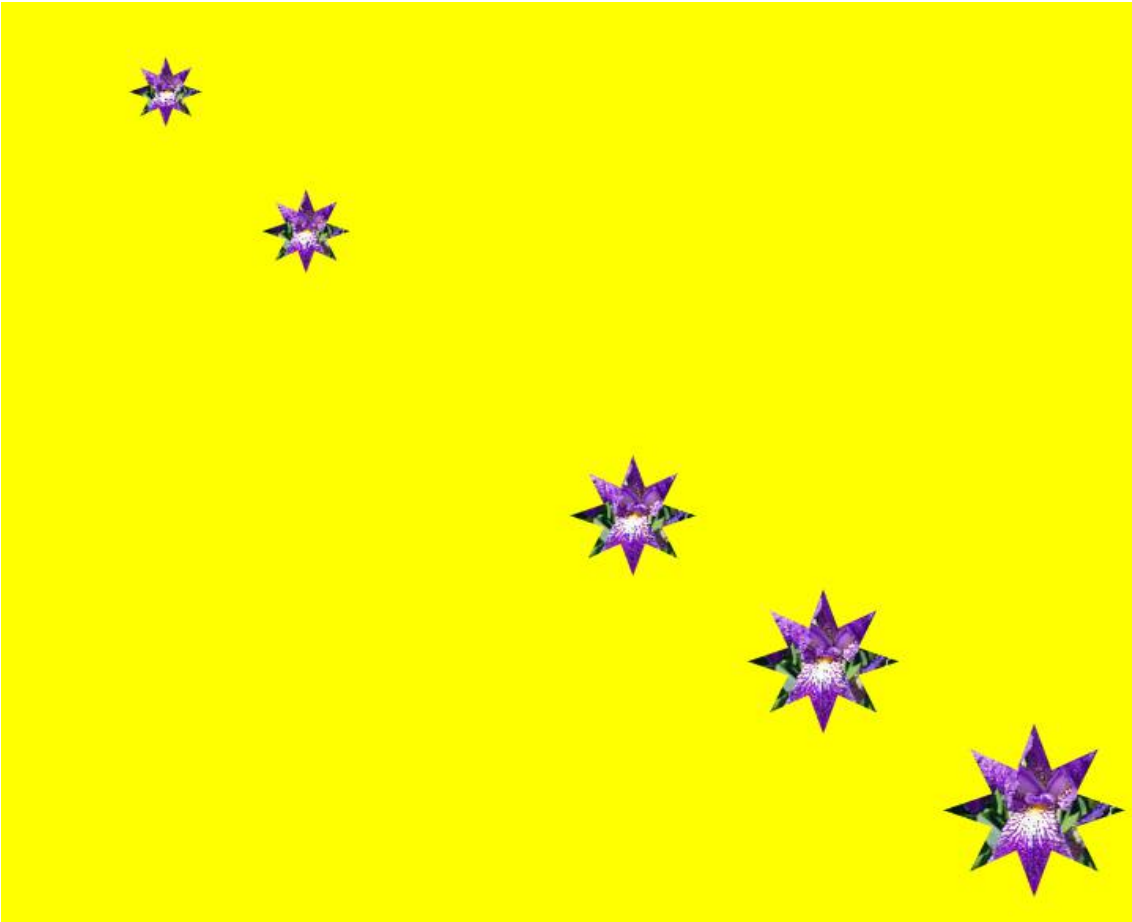
„Fantasijana“ wird **Transparenz** unterstützen, nicht nur bei Farben und Verläufen, sondern auch bei Fotos oder anderen Bildern, die in ein Bild eingefügt werden.

Als vorgefertigte Silhouetten wird „Fantasijana“ nicht nur Verbindungslinien und Pfeile, sondern auch **Skalen, Koordinatensysteme** u.ä. bieten.

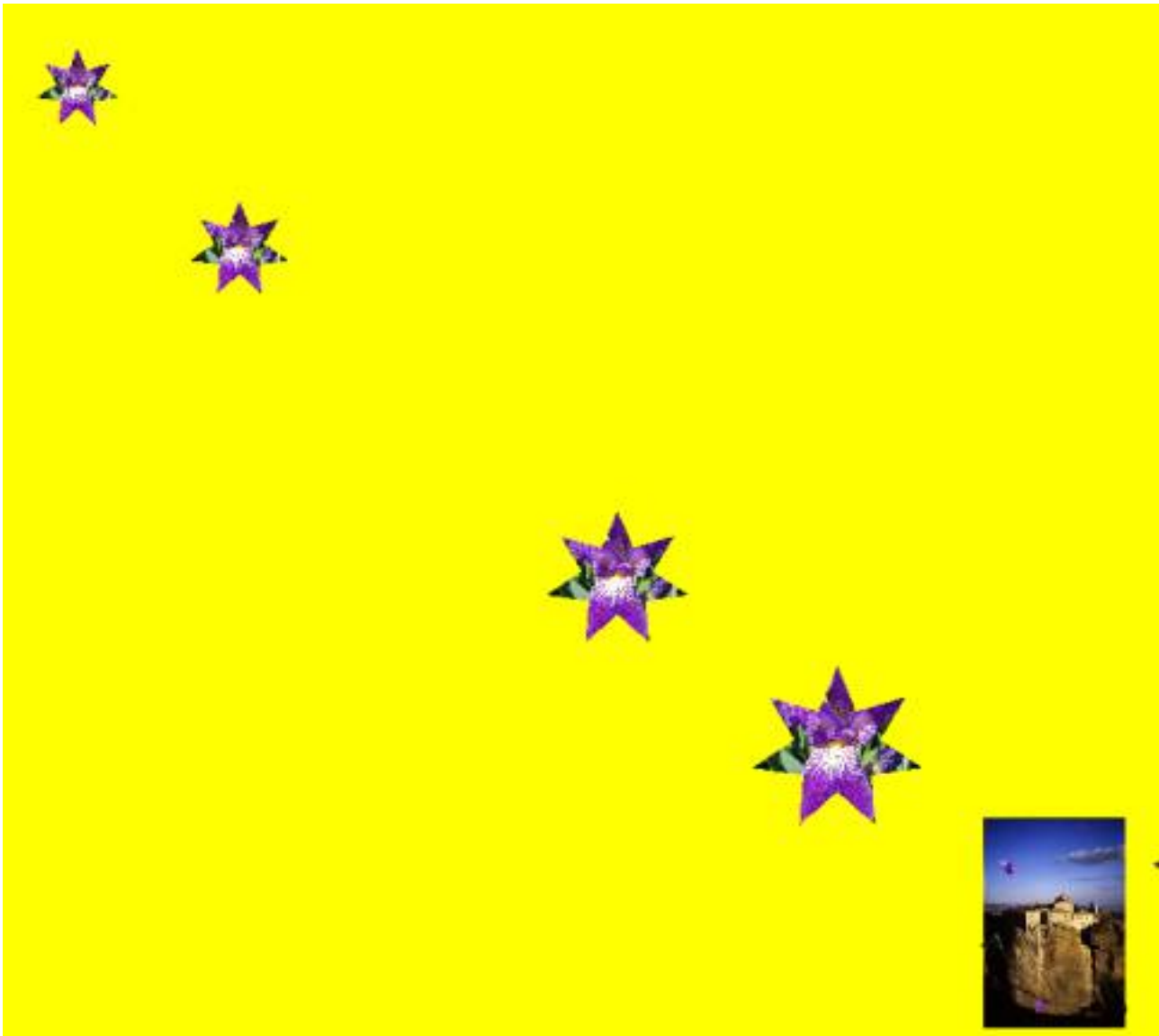
„Fantasijana“ wird **Masken** in flexiblerer Form bieten als beispielsweise SVG. Es wird feste / statische und dynamische Masken geben. Letztere machen „alles“ mit: Vergrößerungen, Verschiebungen, Drehung eines Zeichenobjektes (meist hier: Bildes). Beispiel-Bilder:



*Abbildung 1: Die statische Maske unten rechts macht ein (elliptisch geformtes) Loch in das Foto von der Schwertlilie und lässt das darunter liegende Urlaubsfoto aus Griechenland erkennen.*



*Sternförmige, dynamische Maske*



*Die sternförmige, dynamische Maske und die statische Maske lassen zusammen fast alles von dem erkennen, was darunter liegt*

Die **Unterstützung von Animationen** ist erst angedacht. Langfristiges Ziel ist es dabei, auch nicht-lineare Zusammenhänge animiert darzustellen, beispielsweise freien Fall, Kreisbewegungen, Bevölkerungsexplosion usw. Eine solche **Animation mit nicht-konstanter Geschwindigkeit** wird m.W. auch von der neuesten „Internet-Sprache“ HTML5 (mit Integration von SVG) nicht unterstützt.

Ungefähr ein Dutzend **weitere** bereits getestete und für die Implementierung vorgesehene **Funktionen / Alleinstellungsmerkmale** unterliegen im Detail noch der Geheimhaltung. Dazu gehören Schraffuren, Verzierungen, perspektivisches Verzerren und anderes. Ich bitte zu entschuldigen, dass ich hier nur das Wesentlichste darstellen kann.

**Nochmal zusammengefasst (und um einige Punkte erweitert)** würde das Programm (voll ausgebaut) u.a. folgende Dinge bieten:

1. Volle Funktionalität sowohl eines Objekt-orientierten, wie eines Pixel-orientierten Grafikprogramms
2. Rekursivität:  
Bilder können andere Bilder als Teil enthalten, und dadurch kann das „Teilbild“ an beliebig vielen Stellen in beliebig viele andere Bilder eingebaut werden – entweder genau gleich oder mit kleinen Änderungen  
Erzeugen neuer zu zeichnender Formen aus bestehenden (CAG = Constructive Area Geometry, zu Deutsch: konstruktive Flächengeometrie)
3. Verbindungslinien und Pfeile
4. Bezier-Kurven
5. Transparenzen
6. Verläufe
7. Texturen
8. Filter (Verunschärfen, heller / dunkler, bläulicher etc., schwarz / weiß)
9. Masken / Abdecken eines Teils des Bildes
10. Strichelungen
11. Schraffuren
12. Parkettierungen mit 3-, 4- und (unregelmäßigen) Fünfecken, möglicherweise auch aperiodische (Penrose etc.)
13. Drehungen, Spiegelungen usw. ("Affine Transformationen")
14. Mehrbenutzer-Fähigkeit
15. Versionierung
16. Datenschutz und -sicherheit (Verschlüsselung, Sicherheitskopien, ...)

Diese Liste wirkt zunächst unglaublich lang – und ist noch nicht einmal vollständig. Alles ist jedoch zumindest theoretisch durchdacht; meist sogar getestet. Die Punkte 14 – 16 basieren auf meinen speziellen Berufserfahrungen im Bereich Datenbanken. Bei der Länge dieser Liste bitte ich zu berücksichtigen, dass ich bereits ca. 10 Jahre an der Konzeption von „Fantasijana“ arbeite, wenn auch bisher nur hobbymäßig. Und die Funktionalitäten der Programmiersprache Java sind in diesem Bereich außerordentlich gut.

### **3. Kunden**

#### **1. Schritt**

Schüler, Azubis, Studenten usw. soll ein Basismodul zum Taschengeldpreis angeboten werden. Damit sollen auch rechtliche Probleme durch den Kauf der Software durch Minderjährige ausgeschlossen werden.

Der Funktionsumfang soll sich an den Bedürfnissen dieser Zielgruppe orientieren. Ein Überfrachten mit nicht zielgruppengerechten Funktionen soll vermieden werden. Das „weniger ist mehr“ soll „Fantasijana“ gerade positiv von der Konkurrenz unterscheiden.

Dazu wird auch großes Gewicht auf die Strukturierung gelegt. Durch geschickte Parametrisierung von Funktionen ist nämlich eine Zusammenfassung dieser möglich, ohne gleichzeitig die Gesamtfunktionalität des Programms zu verringern. (Beispiel: Funktion „als Spirale anordnen“ wird als eine Funktion mit Auswahl des Spiralentyps implementiert: Archimedische / „Teppich“, hyperbolische / „Treppenhaus“, logarithmische Spirale / Schnecke usw. Stattdessen könnte man auch für jeden Spiralentyp eine extra Funktion anbieten.)

#### **2. Schritt**

Es ist an die Unterstützung aller geistig Arbeitenden gedacht. Es geht dabei um die Unterstützung der Konzeptions- und Ideenphase: Während CAD-Systeme und beispielsweise Photoshop die Ingenieure, Architekten und Werbefachleute dabei unterstützen, ihre ausgearbeiteten Ideen in die Fertigung bzw. in den Druck zu geben, soll „Fantasijana“ gerade dabei helfen, die Phase davor zu unterstützen: Schnell Entwürfe machen, eine Auswahl daraus treffen, eine Skizze anfertigen, um die Idee(n) mit dem Vorgesetzten oder Kunden besprechen zu können usw.

In diesem Zusammenhang ist es besonders wichtig, die Bedienung des Programms so einfach zu halten, dass die Anwender das Programm nutzen können, ohne dass dabei geistige Kapazitäten für das Programm gebunden werden. Die geistigen Kapazitäten sollen soweit wie möglich für das Problem, die dazugehörigen Lösungsideen und die zugehörige Kommunikation freigehalten werden.

#### **3. Schritt**

Visualisierung von (numerischen) Daten allgemein, Stichworte:

Datenanalyse, Betriebsdaten, wissenschaftliche Untersuchungen – aber mit viel mehr und besseren Möglichkeiten, als das die bisher üblichen Businessgrafik-Programme können.

### **4. Markt und Wettbewerber**

„Fantasijana“ wird eine Mischung aus objektorientierten und Pixeloperationen bieten. Ähnliches gibt es bereits als SVG-Editoren (Inkscape z.B.). Andere Konkurrenten sind nur entweder pixel-

oder objektorientiert (Irfanview, Openoffice-Draw). Die Oberflächen dieser Programme sind allesamt schlecht bis katastrophal oder aber, das Programm ist insgesamt zu groß (Photoshop).

Das Basismodul soll die Konkurrenz insbesondere auf folgenden Gebieten ausstechen:

- ➔ Oberfläche / Bedienung
- ➔ Layout-Funktionen
- ➔ dynamische Masken
- ➔ Transparenz und ähnliches, was bisher (teuren, einarbeitungsintensiven) Profiprodukten vorbehalten war

## 5. Strategie

- Ansprechen junger Kunden, an das Produkt binden
- Niedriger Preis, viele Kunden
- Produkt individualisieren über Splashscreens, die für jede lizenzierte Version extra erzeugt wird. (zumindest teilautomatisiert natürlich)
- Zunächst kostenlose „Softwareprobchen“ mit vollem oder fast vollem Basis-Funktionsumfang verteilen.
- Download über Internet erst im zweiten Schritt; bringt erst dann etwas, wenn ausreichender Bekanntheitsgrad bereits vorhanden. Eine ausreichende Zahl von E-Mail-Adressen müsste sich über eine automatisierte Suche im Internet ermitteln lassen. Kauf wäre bei E-Mail-Adressen von Unternehmen leicht möglich, aber hier ist man wegen der Ansprache von „e.V.s“ oder NGOs etc. wohl zur Selbsthilfe „genötigt“.
- Erweiterungen in Form von Spezialisierungen für einzelne Metiers später zu (geringfügig) höheren Preisen. (Für Ärzte, Rechtsanwälte, Ingenieure etc.)

## 6. Marketing / Werbung

Ein Grafikprogramm – besonders dieses – eignet sich natürlich für jede Form visueller Werbung: Plakat, Bilder im Internet, ... Außerdem ist es geradezu ideal für Preisausschreiben: Wer macht das tollste Bild? Wer hat die besten Tricks auf Lager? Usw.

Dazu kostenlose „Softwareprobchen“ mit vollem oder fast vollem Basis-Funktionsumfang verteilen über E-Mail mit Werbebrief (mit Abbildungen drin).

Da Werbung an Schulen grundsätzlich verboten ist und Ausnahmegenehmigungen teilweise sogar von den einzelnen Schulen erteilt werden müssten, bietet sich folgender Umweg an:

Nicht: Schule → Lehrer → Schüler



Sondern: (Sport-) Verein → Schüler (→ Lehrer → Schüler ... oder Lehrer → Schule → Elternbeirat ... oder ähnlich)

Anmerkung hierzu:

Von den männlichen Schülern ab 15 sind ca. 60% in einem Sportverein, Mädchen entsprechend etwa 40%; insgesamt gibt es in Deutschland weit über eine halbe Million Vereine; ca. 24 Millionen Deutsche sind Mitglied in einem Sportverein.

(Verbreitung über NGOs oder wohltätige Organisationen würde mich auch reizen.)

Weiter: Die Zielgruppe selbst, aber auch der Umweg, auf dem sie angesprochen werden soll, beeinflusst natürlich auch die Funktionalität des Programmes: Es müssen nicht nur die Bedürfnisse der Zielgruppe, sondern auch die des „Umwegs“ berücksichtigt werden. Also z.B. sollte die Erstellung von Vereinszeitungen o.ä. unterstützt werden (evtl. nicht nur einzelne Bilder, sondern Erstellen der Seiten eines „Mitteilungsblättchens“)

## **7. Unternehmen**

Die größte Schwierigkeit bereitet die Beschaffung von gutem Personal in diesem Arbeitsmarkt. Mindestens ein Mitarbeiter erscheint aber erforderlich (durch Zufall vielleicht aber zu beschaffen). Bei der Programmierung soll so gearbeitet werden (so der Code strukturiert werden), dass durch Integration des Programm-Codes mit der Anwender- und Programmierer-Dokumentation automatisch eine einheitliche (und damit effizient einsetzbare) Ordnung entsteht. Das objektorientierte Arbeiten an den Programmen selbst soll also auf den Bereich Dokumentation (Anwender- und Programmierer-Dokumentation!) ausgeweitet werden.

## **8. Finanzen**

Sowohl vom Zeitaufwand, wie auch entsprechend von den Kosten her sind

Softwareentwicklungsprojekte äußerst schwer einzuschätzen. Selbst hochgelobte Methoden wie „function point“ und dergleichen sind – gerade bei einem Entwicklungsprojekt, das gleich in mehreren Bereichen Neues anstrebt – eher als eine Art theoretischer Witz zu sehen, aber für die Praxis völlig untauglich.

[Am Rande: Das liegt überwiegend daran, dass meist erst bei der Implementierung sichtbar wird, welche Detailprobleme sich hinter einem – zunächst auch einkalkulierten – Haupt-Problem verbergen. Das ganze läuft in der Regel rekursiv: Ein zunächst als Detailproblem gesehenes wird bei näherer Betrachtung zum Hauptproblem, zerfällt also dann wieder in eine Anzahl Probleme, deren Schweregrad (sprich: Zeit- und Kostenaufwand) zunächst überhaupt nicht berücksichtigt werden konnte.]

Anhand des bereits Realisierten, der Kenntnis der Entwicklungsumgebung und der Programmiersprache (beides zum Glück äußerst stabil) und der notwendigen Funktionen, die implementiert sein müssen, bevor man sogar mit einer für den Test durch die Kunden geeigneten Version (beta-Version) an den Markt gehen kann, wage ich folgende Abschätzung: Minimal 3 Personenmonate, wahrscheinlich ein halbes Jahr, wenn es ganz schlecht läuft: 2 Jahre. (Eine andere Daumenregel kommt auf 1 Personenjahr: Hohler Bauch = 1 Monat Arbeit, das mal  $\pi^2$  ... also die Erfahrung sagt: so knapp 1 Jahr.)

Sofern ich ohne einen Gehilfen arbeiten würde, wären € 50.000 / Jahr erforderlich. (Die relativ geringe Summe ergibt sich aus Corona + keine Lust, COBOL auf Java zu portieren + keine Lust auf Chef, der nur ständig nervt und einen an der Arbeit hindert usw.)

Mit Hilfe das Doppelte, spätere Werbemaßnahmen etc., wenn die Vermarktung beginnt: € 100.000. In der Summe müsste man mit der für Softwareentwicklungsprojekte geradezu lächerlichen Summe von € 300.000 in den ersten zwei Jahren (also bis spätestens der erste nennenswerte Umsatz erzielt wird) hinkommen. Kosten für Büro etc. spielen keine bzw. eine nur völlig untergeordnete Rolle. Java-Entwicklungen sind praktisch kostenlos, was Compiler und so etwas angeht.

Geplant ist die Vermarktung mindestens einer Low-Price-Version. Ein Preis von ca. 10 – 20 Euro pro Lizenz soll hohe Verkaufszahlen bei niedrigem Anteil Schwarzkopien garantieren,. Außerdem sollen dadurch juristische Probleme umgangen werden („Taschengeldparagraph“). Da die gesamte Zielgruppe mehrere Millionen Personen umfasst (ca. ½ Million je Jahrgang im Alter von 15 – 25 bilden den Kern der Zielgruppe), müsste sich mit dieser Strategie ein ausreichender Umsatz generieren lassen. (Umsatz = Anzahl Lizenzen \* Stückpreis; 90% Leute, die kein Interesse haben oder grundsätzlich nur mit Schwarzkopien arbeiten (Zum Vergleich: Laut einer Umfrage im Jahr 2007 mit allerdings nur 500 Teilnehmern benutzen 58 % der Photoshop-Benutzer das Programm als Schwarzkopie. Im gleichen Jahr war Photoshop die vierthäufigste schwarzkopierte Software. - vgl. [https://de.wikipedia.org/wiki/Adobe\\_Photoshop](https://de.wikipedia.org/wiki/Adobe_Photoshop)); je 5% Marktabdeckung jeweils in den ersten zwei Jahren mit Umsatz = 50.000 Lizenzen je € 10,--, macht bei vernachlässigbaren Vertriebskosten ungefähr € 500.000 ROI. Mehr als diese Überschlagsrechnung erscheint nicht möglich, und die sagt: Das Risiko ist bereits dann vertretbar, wenn man nur von deutschen Kunden ausgeht und die langfristigen Chancen (Erweiterungen, „professional Edition“, individuelle Ergänzungen, ... ) zunächst völlig außer acht lässt. (Zu der Schätzung bzgl. der Anzahl der Lizenzen folgende Vergleichszahlen: Das relativ unbekannte, kostenlose DTP-Programm „Scribus“ wurde in ca. 20 Jahren über 200.000 mal heruntergeladen. Das bekannte pixel-orientierte Programm „Photoshop“ kommt nach 30 Jahren auf ca. 10 Millionen Nutzer weltweit.)

## 9. SWOT-Analyse

S:

- Ansprache Jugendlicher als Zielgruppe: damit langfristige Kundenbindung möglich
- legale Ansprache der Zielgruppe über Vereine (Sport, Musik, ...) möglich
- „Zurückschlagen“ (Einbauen stets neuer Funktionen) bei Nachziehen der Konkurrenz bereits vorgeplant
- sehr durchdachte Programm- und Oberflächenstruktur von Vorneherein --> geringe Aufwände für Fehlerbehebung und Einbau zusätzlicher Funktionen zu erwarten --> niedrige Wartungskosten langfristig
- Diversifikation durch Anpassung (notfalls auch an ganz spezielle Anforderungen) / Erweiterung des Programms vorgeplant --> Es gibt Ausweichmöglichkeiten selbst bei übermächtiger Konkurrenz
- Umsatzgenerierung auch über „Nebenprodukte“ möglich (Bücher zum Thema, Libraries für andere Softwarehäuser z.B.)
- Zahl der Schwarzkopien senkbar durch Individualisierung des Programms (Splashscreens, Framehintergründe, Defaults für Bilder ...)

W:

Generelle Unsicherheiten:

Neuartiges Produkt

Neue Zielgruppe

Softwareentwicklung

O:

Zunehmendes Lernen auch über das Internet, auch wenn „Corona“ nicht mehr aktuell

T:

Ungewöhnlich hohe Zahl von Schwarzkopien (> 90 %)

Generell geringes Interesse (z.B. wegen drastischer, allgemeiner Wirtschaftskrise)

## 10. Meilensteine

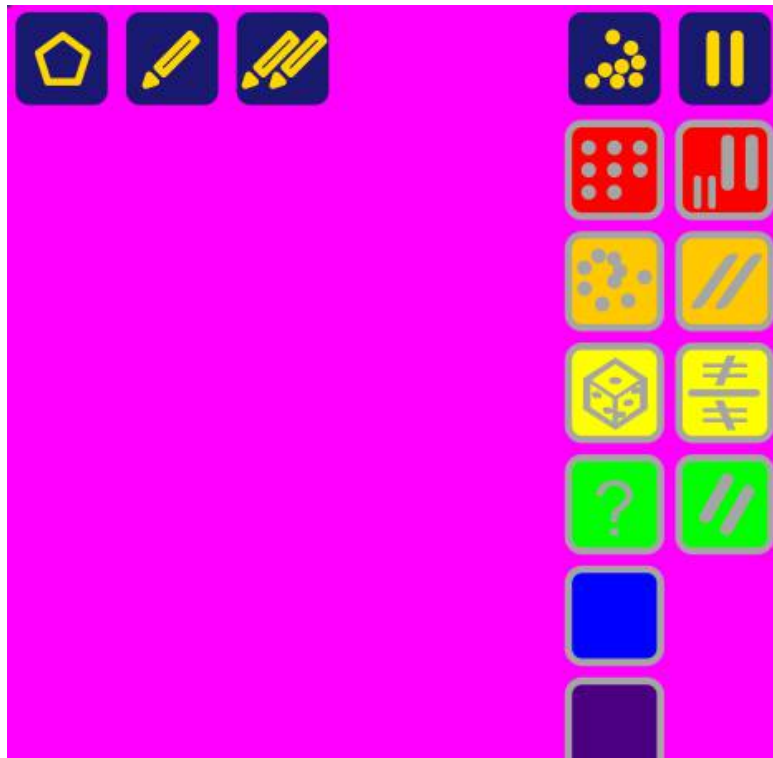
Überarbeitung der Datenstrukturen

Fertigstellen (Implementieren) der Oberfläche

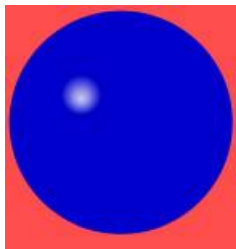
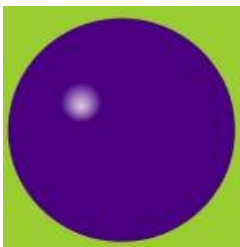
Implentierung allgemeiner Funktionen (logging, Undo, ...)

**Anhang: Beispielbilder, erzeugt mit dem Prototypen**

In Arbeit befindlicher Entwurf für das „Menü“ aus Symbolleisten:



Glanzpunkt – gesetzt mit Hilfe eines Verlaufs von Weiß nach Transparent



Rebsorten, beschriftet (Teile des Bildes, die Schleichwerbung für ein bestimmtes Weinhaus wären, wurden unkenntlich gemacht.)



Gruß zum Jahresende 2020



Spirale mit einer Folge von Urlaubsfotos von einem Sonnenaufgang über dem Mittelmeer  
(entstanden am 4.1.2013 zwischen halb acht und acht; verkleinert ca. 10x):



Plan für das Stellen von Möbeln in unserem neuen Haus auf Basis der Zeichnung des Architekten:

