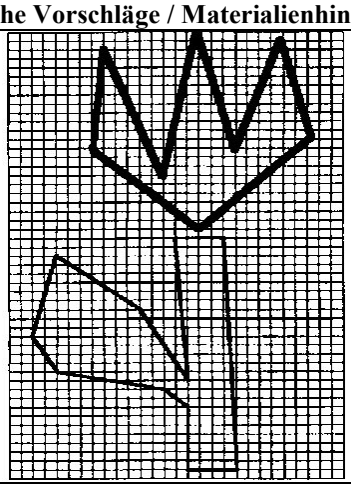
Thema: Verarbeitung von Bildern mit Informatiksystemen

Rahmenbedingungen: Durchführung z.B. in Jahrgangsstufe 9.2,
Thema in 9.1: Information und Kommunikation im Internet
Einfaches pixelorientiertes Grafikprogramm (z.B. MS-Paint-Windows98)
LOGO-Programmierumgebung (z.B. MSW-Logo 3.6 [1])
Vektororientiertes Grafikprogramm (z.B. MS-Word-Grafik)

Zeitvolumen: maximal ein Halbjahr mit 3 Wochenstunden

Unterthemen und Inhalte	**Methodische Vorschläge / Materialienhinweise**
Bedeutung und Anwendung von Computergrafiken - Kunst, Gebrauchsgrafik, Grafikanalysen in der Astronomie, Computertomographie - zweidimensional/dreidimensionale Computergrafik - Anzeige bei GPS-Systemen im Auto, Torten- oder Säulendiagramme der Statistiker, grafische Benutzeroberflächen, Computerspiele - Geräte zur Eingabe, Verarbeitung und Ausgabe von Computergrafiken (EVA) - Bildpunkte (Pixel), Auflösungsvermögen (dpi)	Im Unterrichtsgespräch werden, unterstützt durch Beispiele auf Folien, Anwendungsgebiete von Computergrafiken und verschiedene Typen von Computergrafiken aufgezeigt (z.B. schematische und realitätsnahe, 2D und 3D, statische und dynamische Abbildungen). Das aus dem 1. Halbjahr bekannte EVA-Prinzip wird auf die Computergrafik angewandt. Im Zusammenhang mit der Nennung von Grafikkarte oder Monitor werden die Begriffe Bildpunkte (Pixel), Auflösung und Auflösungsvermögen (dpi) angewandt.
Ein Bild vom Nikolaushaus mit Garten - pixelorientiertes Grafikprogramm - objektorientierte Werkzeuge (Linie, Stift, ...) - Attribute zu Objekten (Stiftgrösse, ...) - Piktogramme zur Mensch-Maschine-Kommunikation und andere Steuerungselemente - Zoomen und der Zoomfaktor als Verhältnis zwischen Originalgröße und Ausschnitt des Bildes - Treppeneffekt in Pixelgrafiken (Informationsverlust) - Kopieren, Verschieben, Ausschneiden und Löschen von ausgewählten Bildbereichen	Mit einem pixelorientierten, möglichst einfach gehaltenen Grafikprogramm wird eine Grafik mit dem Haus vom Nikolaus und zugehörigem Blumengarten erstellt. Dabei erfahren die Schüler den Umgang mit einer menüorientierten Oberfläche, erkunden die Werkzeugleiste und arbeiten mit unterschiedlichen grafischen Objekten. Zur Gestaltung kleiner Objekte wird die ZOOM-Funktion verwendet, wobei der Treppeneffekt bei Pixelgrafiken deutlich wird. Die Erfahrungen zu den Funktionen und zum Aufbau eines Grafikprogramms werden gesammelt und thematisiert.
Digitalisieren von Bildern - Rechner benötigen Zahlen zur Verarbeitung - Digitalisierung – Information als Zahlen darstellen - Digitalisieren einer s/w-Zeichnung (Tulpe) - Pixelorientierte und vektororientierte Repräsentation mit Zahlenpaaren - Informationsverlust beim Digitalisieren - Datenvolumen contra Informationsverlust	Auf einer Folie wird mit einer Darstellung einer "Black Box" der Übergang von der Anwendungsebene zur Ebene der Auseinandersetzung mit der internen Funktionsweise einer Grafiksoftware symbolisiert und verbal motiviert. Auf eine Zeichnung (Arbeitsblatt einer Tulpe wird ein transparentes Millimeterpapier mit einem Koordinatensystem gelegt (s.u.). Die Schüler überlegen, wie die Bildinformation in Zahlen repräsentiert werden kann, zeichnen entsprechend auf der Folie, ermitteln die zugehörigen Zahlen und stellen ihre Ergebnisse vor (gegebenenfalls ergänzt der Lehrer die vektorielle bzw. pixelorientierte Repräsentatonsform). Anhand eines Vergleichs der Originalzeichnung mit dem pixel- bzw. vektororientierten Repräsentationsmodell werden Informationsverlust und Aufwand gegenübergestellt.

[1] Es ist darauf zu achten, dass unter Windows das Arbeitsverzeichnis im MSW-LOGO-Piktogramm auf das von den Schülern zu benutzende Arbeitsverzeichnis eingestellt ist. Zudem muss sich das Verzeichnis LOGOLIB im!!! Verzeichnis LOGO befinden!

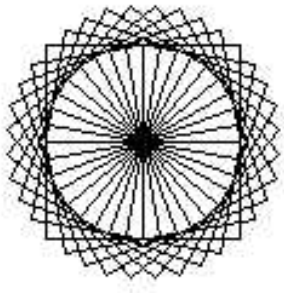
Unterthemen und Inhalte	**Methodische Vorschläge / Materialienhinweise**
	
Verwalten der Daten in einer Datei - Speichern der Zahlenpaare mit einem Texteditor - Zugriff auf Datensätze in sequentiellen Dateien - Steuerung des Ein-/Ausgabestroms - Anweisungen der Programmiersprache für den Dateizugriff	Mit einem einfachen Texteditor werden die ermittelten Zahlenpaare in eine Datei geschrieben, wobei jedes Zahlenpaar in eine Zeile geschrieben wird (Datensatz). Aufgrund des geringeren Datenvolumens wird den Schülern nahegelegt, die vektoriellen Daten zu verwenden. Das Prinzip sequentieller Dateien („Sichtfenster“ läuft nur in einer Richtung über die Datensätze“) wird erläutert. Dabei werden schon die entsprechenden Anweisungen der Programmiersprache (hier: LOGO) eingeführt: openread, openwrite, setread, setwrite, readlist, close, eofp.
Einführung in die Programmiersprache/-umgebung - Prinzip der Turtlegrafik - Anweisungen mit Parametern zur Steuerung der Turtle - Verwendung des Begriffs Algorithmus (Befehlsabläufe) - Formulierung einfacher Algorithmen - Implementieren der Algorithmen als Prozeduren - Speichern und Laden von Programmen in der Programmierumgebung	Da Logo noch nicht im Unterricht verwendet wurde, wird das Prinzip der Turtlegrafik erläutert, indem ein Schüler die Schildkröte spielt und ein anderer Schüler Anweisungen aus einem Anweisungskatalog erteilt (forward N, right N, left N, **setpos** [0 0] , pendown, penup, repeat N [<...>]), um Quadrat, Dreieck, u.a. zu zeichnen. Je nach Wetterlage findet das Spiel auf sandigem Boden oder Papier statt. Die Schüler notieren die verwendeten Befehlsabläufe. Diese werden im Unterrichtsgespräch besprochen. Nach einer kurzen Einführung in die Programmierumgebung implementieren die Schüler die einzelnen Algorithmen als Prozeduren in LOGO.

Der Computer malt nach Zahlen
- Schleife mit Abbruchbedingung im Kopf
- Algorithmus zum Lesen von Datensätzen aus einer Datei und deren graphische Interpretation
Bsp.
`to vektorgrafik_malen`
`openread "blume.vec`
`setread "blume.vec`
 	
`pendown`
`while [not eofp] [setpos readlist]`
`penup`
 	
`setread []`
`close "blume.vec`
`end`
	Die eigentliche Problemstellung wird neu motiviert. Im Unterrichtsgespräch wird ein Algorithmus in der Programmiersprache entwickelt, der aus der Datei mit den Zahlenpaaren jeweils einen Datensatz ausliest und die Turtle entsprechend zeichnen lässt.
	Die Schüler implementieren den Algorithmus und testen ihn an ihren Dateien mit den Zahlenpaaren.
	
	Anmerkung:
	Es ist sicherzustellen, dass die Schüler ohne Setzen des Arbeitsverzeichnisses in dem Programm auf die Dateien mit den Bilddaten zugreifen können, d.h. den Schülern wird mitgeteilt, wo sie ihre Dateien ablegen müssen. Am Besten werden die wichtigsten Rahmenbedingungen zu der Programmiersprache auf einem Arbeitsblatt bereitgestellt. Mit `zoom` N kann das schon bekannte Zoomen auch in LOGO erfolgen.

Unterthemen und Inhalte	**Methodische Vorschläge / Materialienhinweise**
Verbesserung der Bildrepräsentation - bedingte Verzweigung - Farbtiefe – Anzahl der Farben - Additive und subtraktive Farbmischung - RGB- und CMYK-Modell Beispiel mit Datei ... `while [not eofp] [ifelse readlist=[1] [pendown] [penup] setpencolor readlist setpos readlist]`	Das Datenmodell der Bilder berücksichtigt gegebenenfalls noch nicht, dass der Stift gelegentlich abgesetzt werden muss (0 bzw. 1 als Kodierung). Der Kodewert wird vor jeder Linie in einer separaten Zeile gespeichert. Die Abfrage erfolgt im Programm mit einer bedingten Verzweigung, die zuerst umgangssprachlich von den Schülern formuliert wird. Die Schüler suchen im (elektronischen) Handbuch der Programmiersprache die Anweisung zum Setzen der Stiftfarbe. Im Unterrichtsgespräch werden die Varianten der Farbmischung besprochen; beim Licht (additiv, RGB)und der Malerei (subtraktiv, CYMK). Der Farbwert wird in einer Zeile zwischen Stift-Kodewert und den Koordinaten abgelegt, um ohne Variablen auszukommen. Auf die Behandlung der Datenkompression wird an dieser Stelle verzichtet.
Ein vektororientiertes Bild vom Nikolaushaus - vektororientiertes Grafikprogramm - objektorientierte Werkzeuge (Linie, Stift, ...) - Attribute zu Objekten (Stiftgrösse, ...) - Piktogramme zur Mensch-Maschine-Kommunikation und andere Steuerungselemente - Zoomen und der Zoomfaktor als Verhältnis zwischen Originalgröße und Ausschnitt des Bildes - Skalierung von Vektorgrafiken (kein Informationsverlust, vgl. Pixelgrafik) - Kopieren, Verschieben, Ausschneiden und Löschen von ausgewählten Bildteilobjekten - Umwandeln von Pixelgrafiken in Vektorgrafiken und umgekehrt (Informationsverlust)	Mit einem vektororientierten, möglichst einfach gehaltenen Grafikprogramm wird eine Grafik mit dem Haus vom Nikolaus und zugehörigem Blumengarten erstellt. Dabei erkunden die Schüler die Werkzeugleiste und arbeiten mit unterschiedlichen grafischen Objekten. Zur Gestaltung kleiner Objekte wird die ZOOM-Funktion verwendet, wobei die Skalierung von Vektorgrafiken deutlich wird. Die Erfahrungen zu den Funktionen und zum Aufbau eines Grafikprogramms werden gesammelt und mit denen vom pixelorientierten Grafikprogramm verglichen.
Computerkunst – Verfremden von Fotos - Digitalisieren mit Hilfe eines Scanners - Operationen auf Pixelmustern - Bildfälschung, Urheberrechte	Ein Foto wird eingescannt, Scannen thematisiert, das entsprechende Grafikprogramm ausgewählt und einfache Funktionen zu künstlerischen Gestaltung angewandt (Ausschneiden, Drehen, Spiegeln, Füllen,..) Anhand einiger Beispiele werden Bildfälschung („Kohl küsst Madonna“) und Urheberrecht („Laden von Bildern aus dem Web“) thematisiert.
Computerkunst – Grafiken erzeugen lassen - Zählschleife - Variablen (call by value) - Prozeduren mit Parametern - Kommentare - Problem numerischer Näherungen (Kreise) - Rekursion und Iteration am Beispiel einfacher und selbstähnlicher Objekte (z.B. Sierpinski-Dreieck)	Verschiedene Grafiken (s.u.), die mit der Turtle erzeugbar sind und auf der wiederholten Zeichnung ähnlicher Objekte beruhen, werden schrittweise den Schülern auf Folie präsentiert, auf ihre Basisobjekte und ihre Erzeugung aus diesen untersucht und mit LOGO erzeugt. Dabei werden schrittweise weitere Programmiersprachenkonstrukte notwendig und eingeführt (siehe Dümmler-Lehrbuch). Bei der Einführung eines Variablenmodells wird auf den Vergleich mit Speicherzellen/Schubladen verzichtet, um den funktionalen Charakter von LOGO zu erhalten, der ein rekursives Arbeiten erforderlich macht. Stattdessen wird korrekter von "textueller Ersetzung" gesprochen.
Weitere Unterthemen (je nach verfügbarer Zeit) - Erfassen eines Bildes mit der Maus in LOGO - Grafiktypen in Office-Programmen und ihre jeweiligen Anwendungsgebiete - Veränderung der Arbeitswelt durch die Automatisierung grafikverarbeitender Vorgänge	

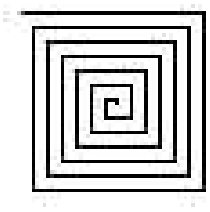
Literatur:
Baues, J. et al.: „Informatik erleben – Lehr- und Übungsbuch“. 2 Bände, Dümmler1996
Engelmann, Lutz (Hrsg.): „Informatik – Informationstechnische Grundbildung“. Paetec 2000
http://didaktik.cs.uni-potsdam.de/HyFISCH/Informieren/computergrafik

Beispielgrafiken zur Einführung neuer Programmierprinzipien:



```
to rechteck
pendown
repeat 4 [forward 40 right 90]
penup
end

to rosette
pendown
repeat 36 [rechteck right 10]
penup
end
```



```
to rechteckspirale :laenge
pendown
forward :laenge
left 90
if (:laenge > 2) [rechteckspirale (:laenge - 2) ]
penup
end
```