

**Mecklenburg
Vorpommern**



Institut für Qualitätsentwicklung

**Universität
Rostock**



Traditio et Innovatio



Informatische Bildung in Mecklenburg-Vorpommern

Mecklenburg-Vorpommern

Spiele entwickeln und multimedial dokumentieren (Klasse 7)

Lutz Hellmig
Universität Rostock
Institut für Informatik

Gewusst wo: <https://elearn.bildung-mv.de>

- ▶ Startseite
 - ▶ Kursbereiche
 - ▶ Gegenstandsbereiche des Unterrichts
 - ▶ Informatik
 - ▶ Informatik und Medienbildung

Spiele entwickeln und multimedial dokumentieren

ca. 10 h

Auf den Erfahrungen aus der Jahrgangsstufe 6 aufbauend entwickeln die Schülerinnen und Schüler erstmalig systematisch ein interaktives Spiel in einer blockbasierten Entwicklungsumgebung. Sie festigen ihre Kompetenzen im Algorithmieren und wenden ihre bisher erworbenen Fähigkeiten beim Dokumentieren mithilfe digitaler Medien an.

7.1 Spiele entwickeln und multimedial dokumentieren

- Algorithmen beschreiben, interpretieren und modellieren (incl. Fehlersuche- und Behandlung)
- Schachtelstrukturen und zusammengesetzte Bedingungen verwenden
- Operationen auf numerischen Daten, Zeichenketten, Booleschen Werten
- Vektor- und Rastergrafiken für das Spiel gestalten
- Dokumentation erstellen (incl. Textproduktion)
- Urheberrecht beachten
- Komprimierung exemplarisch erläutern (Bildcodierung)

Vorschläge zur Gliederung der Stoffeinheit

Vorschlag 1

- 2 Spiele analysieren und korrigieren
- 6 Eigenes Spiel entwickeln
- 2 Spiel multimedial dokumentieren
- 1 Auswertung und Präsentation
- 1 Spiele reflektieren, Komprimierungsverfahren erleben

Vorschlag 2

- 2 Spiele analysieren und korrigieren
- 6 Eigenes Spiel entwickeln und dokumentieren
- 2 Auswertung und Reflexion
- 2 Komprimierungsverfahren erleben

7.1 Spiele entwickeln und multimedial dokumentieren

- Algorithmen beschreiben, interpretieren und modellieren (incl. Fehlersuche- und Behandlung)
- Schachtelstrukturen und zusammengesetzte Bedingungen verwenden
- Operationen auf numerischen Daten, Zeichenketten, Booleschen Werten
- Vektor- und Rastergrafiken für das Spiel gestalten
- Dokumentation erstellen (incl. Textproduktion)
- Urheberrecht beachten
- Komprimierung exemplarisch erläutern (Bildcodierung)

6.1 Entscheidungen und Spiele

- Variablen
- Boolesche Daten
- Bedingungen in Schleifen und Verzweigungen
- Logische Operationen
- Fehler systematisch suchen
- ggf. Schachtelung

6.3 Texte

- Formatvorlagen nutzen
- Bilder einfügen und anpassen

5.1 Programmieren

- Sequenzen
- Wiederholung mit fester Anzahl
- Algorithmen beschreiben, testen
- Einfluss von Parametern

5.2 Bild und Grafik

- Raster- und Vektorgrafik
- Farbcodierung

5.3 Präsentationen

- Bilder einfügen und anpassen
- Urheberrecht beachten

7.1 Spiele entwickeln und multimedial dokumentieren

- Algorithmen beschreiben, interpretieren und modellieren (incl. Fehlersuche- und Behandlung)
- Schachtelstrukturen und zusammengesetzte Bedingungen verwenden
- Operationen auf numerischen Daten, Zeichenketten, Booleschen Werten
- Vektor- und Rastergrafiken für das Spiel gestalten
- Dokumentation erstellen (incl. Textproduktion)
- Urheberrecht beachten
- Komprimierung exemplarisch erläutern (Bildcodierung)

6.1 Entscheidungen und Spiele

- Variablen
- Boolesche Daten
- Bedingungen in Schleifen und Verzweigungen
- Logische Operationen
- Fehler systematisch suchen
- ggf. Schachtelung

6.3 Texte

- Formatvorlagen nutzen
- Bilder einfügen und anpassen

5.1 Programmieren

- Sequenzen
- Wiederholung mit fester Anzahl
- Algorithmen beschreiben, testen
- Einfluss von Parametern

5.2 Bild und Grafik

- Raster- und Vektorgrafik
- Farbcodierung

5.3 Präsentationen

- Bilder einfügen und anpassen
- Urheberrecht beachten

Schwerpunkte

1. Spielentwicklung*

- Operieren mit Zeichenketten
- Projektplanung und -entwicklung
- Möglichkeiten der Dokumentation und Präsentation

2. Lauflängenkompprimierung

* Die Programmierprojekte sind in Scratch und Snap! realisierbar.



Zeichenketten

Algorithmus (Skript)

Ein Algorithmus ist eine Handlungsvorschrift mit einer eindeutigen und endlichen Abfolge von elementaren Anweisungen, Schleifen und Verzweigungen, die auch ineinander geschachtelt sein können.

Befehl (elementare Anweisung)

Ein verständlicher, grundlegender und absolut eindeutiger Befehl sagt einem Objekt, was es zu tun hat.



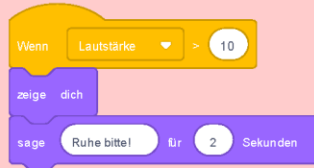
Sequenz (Abfolge)

Die Nacheinanderausführung von Befehlen muss systematisch geplant und durchdacht werden. Ein Fehler in der Reihenfolge führt zu einem anderen Verhalten des Objekts.



Ereignis

Programme können auf Ereignisse reagieren und dann ein Skript starten, Ereignisse sind zum Beispiel das Drücken einer Taste oder das Empfangen eines Geräuschs.



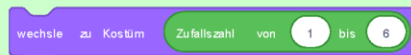
Parameter

Viele Befehle lassen sich durch Parameter steuern. Man kann zum Beispiel angeben, wie weit eine Figur laufen, wie lange eine Pause dauern oder mit welcher Lautstärke ein Klang abgespielt werden soll.



Zufall

Computer können zufällige Zahlen erzeugen. Damit lassen sich unvorhersehbare (zufällige) Dinge in einem Programm realisieren.



Wiederholung (Schleife)

Befehle lassen sich eine bestimmte Anzahl, fortlaufend (bis zum Schließen des Programms) oder bis eine bestimmte Bedingung erfüllt ist, wiederholen.



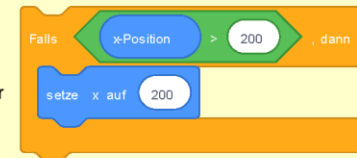
Bedingung

Bedingungen sind immer *wahr* oder *nicht wahr*. Sie lassen sich durch Vergleiche bilden. Durch die Verknüpfung mit den Operatoren UND, ODER und NICHT entstehen neue Bedingungen. Bedingungen werden in sechseckigen Blöcken dargestellt.



Verzweigung (Auswahl)

In Abhängigkeit von einer Bedingung kann ein Programm entscheiden, ob eine Befehlsfolge abgearbeitet wird oder nicht.



Variable

Eine Variable kann man sich als eine beschriftete Schachtel vorstellen. In der Schachtel wird eine Information (eine Zahl, eine Zeichenkette, ein Wahrheitswert, ...) abgelegt. Sie kann zu einem späteren Zeitpunkt gelesen und verändert werden.



Operator

Berechnungen, logische Verknüpfungen, arithmetische Vergleiche und die Ermittlung von Daten aus Zeichenketten sind Beispiele für Operationen auf Daten, die mithilfe von Operatoren realisiert werden können.



Zeichenkette

Beim Programmieren wird Text als lange Folge einzelner Zeichen definiert. Die Zeichen der Zeichenkette werden intern durchnummeriert. Jedes Zeichen lässt sich einzeln lesen und schreiben. Zeichenketten können zusammengefügt werden, um neue Zeichenketten zu erzeugen.



Operieren mit Zeichenketten

1. Probiere das Programm Städteraten MV aus.

<https://scratch.mit.edu/projects/469872058>

2. Schaue in das Programm hinein und füge die vorbereitete

Anweisung



an die am besten geeignete Stelle im Programm ein.

3. Probiere das Programm erneut aus.

Erkennt Du das Problem? Kannst Du es lösen?

Programmerweiterungen

<https://scratch.mit.edu/studios/28464159/>



Wenn ich Nachricht1 empfangen

sage verbinde verbinde Das Lösungswort hat und Länge von Lösung und Buchstaben für 2 Sekunden



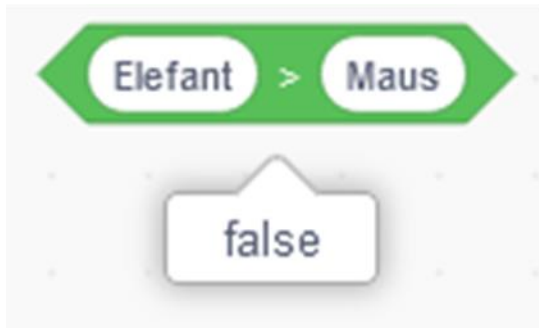
Wenn ich nachricht2 empfangen

setze stelle auf Zufallszahl von 1 bis Länge von Lösung

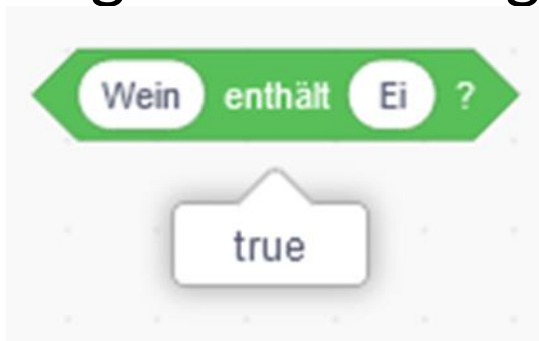
sage verbinde verbinde Der und stelle und verbinde . Buchstabe ist und Zeichen stelle von Lösung für 2 Sekunden

Operationen mit Zeichenketten

- Vergleichsoperationen

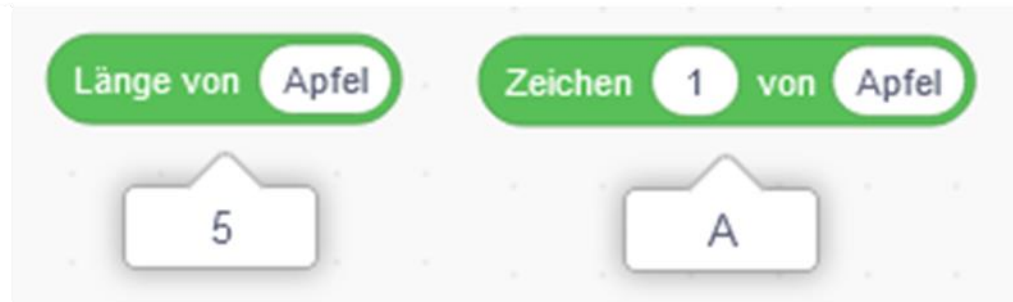


- Prüfung auf Substrings



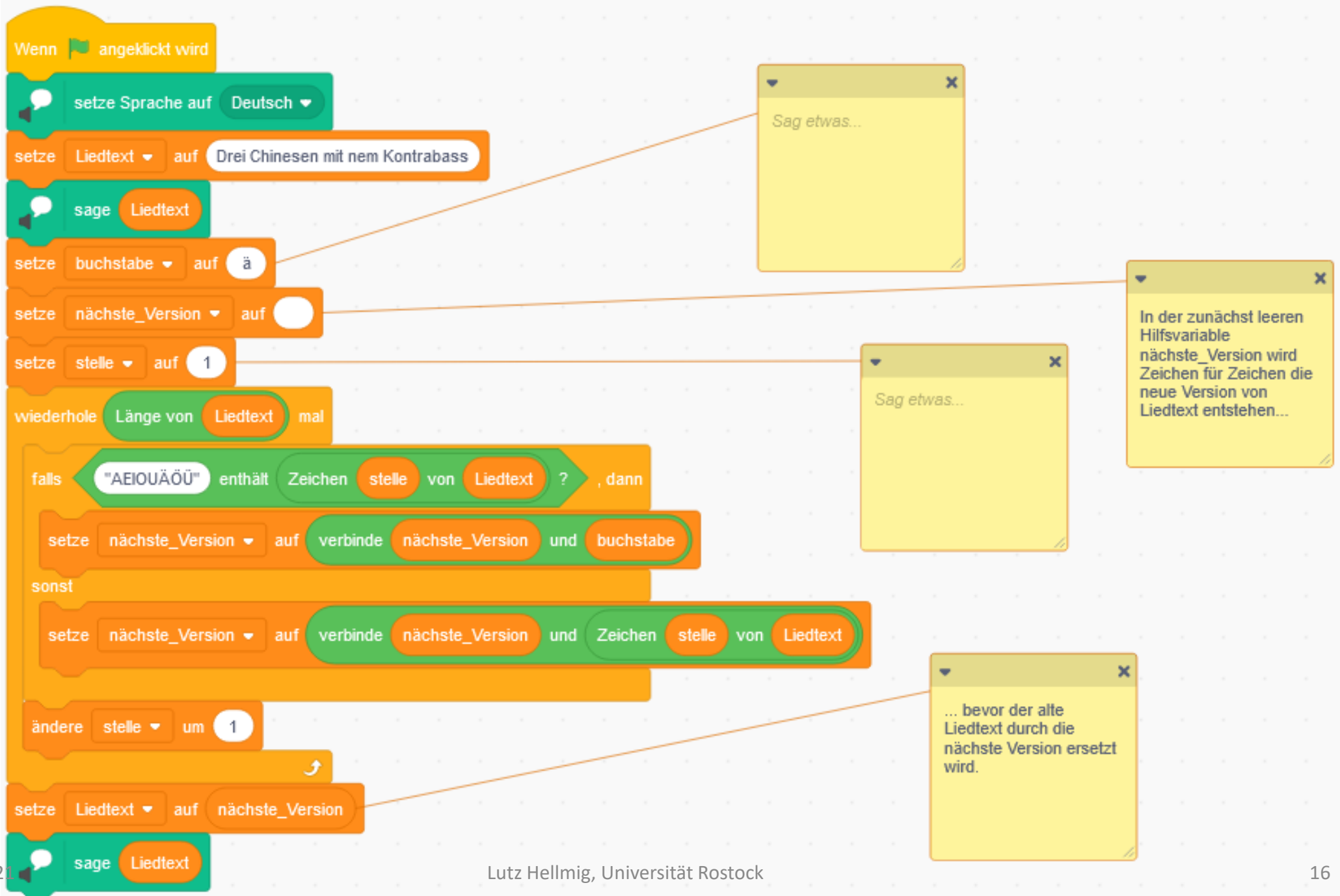
Operationen mit Zeichenketten

- Lesen



- Verknüpfen





Anwendungen entwickeln

Benutzen-Analysieren-Gestalten-Verankern

1. Benutzen

- Fehlerhafte oder unvollständige Informatiksysteme benutzen und Probleme entdecken

2. Analysieren

- Fehlerursachen anhand des Systems erkunden
- Erweiterungsmöglichkeiten mit den gegebenen Mitteln erkennen

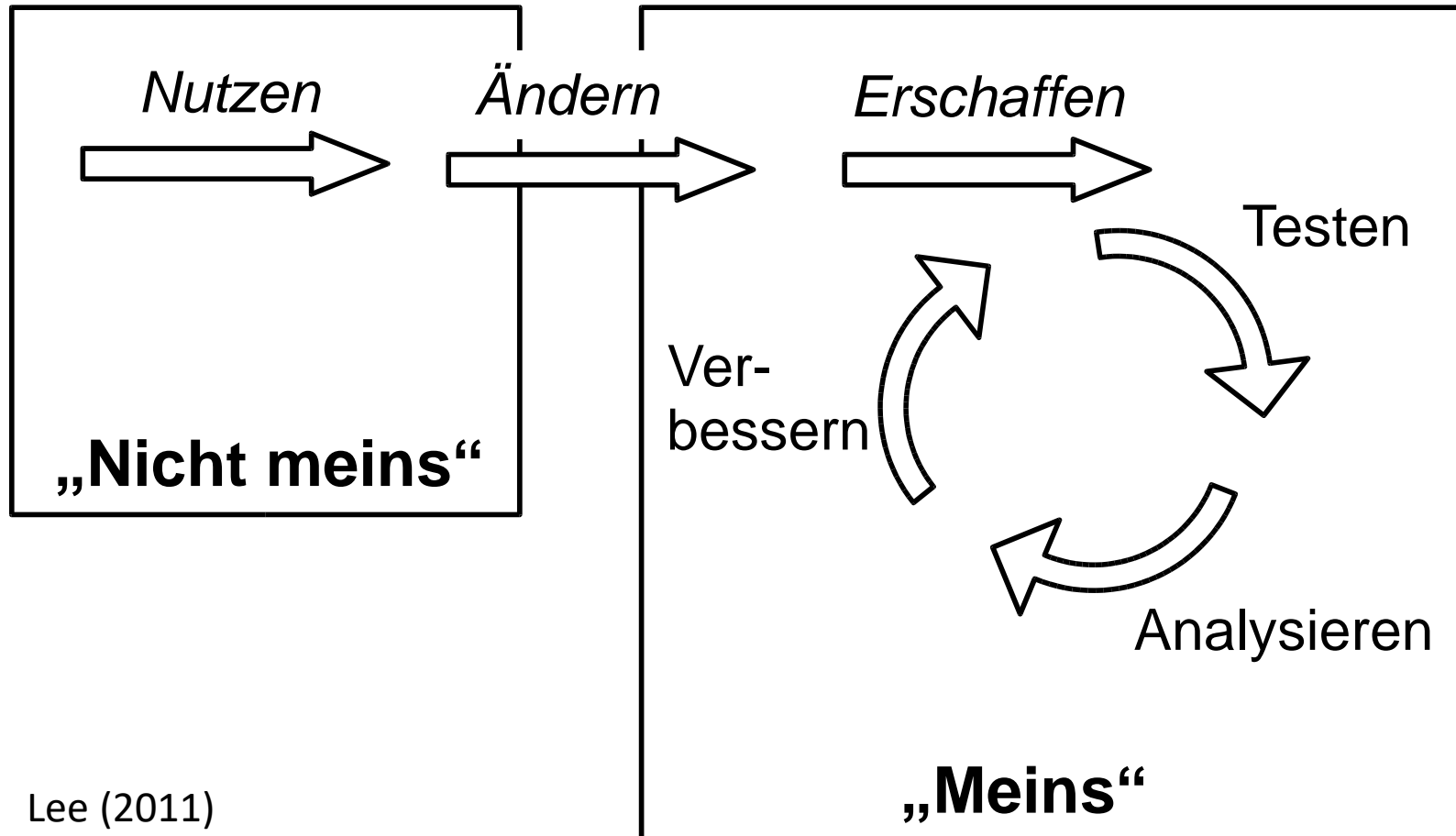
3. Gestalten

- Fehler beheben
- das System schrittweise erweitern (durch zunehmend offene Aufgaben)

4. Verankern

- sich des Gelernten bewusst werden
- vom Beispiel abstrahieren

Ein eigenes Projekt entwickeln



Analysieren und Gestalten – Aufgabentypen

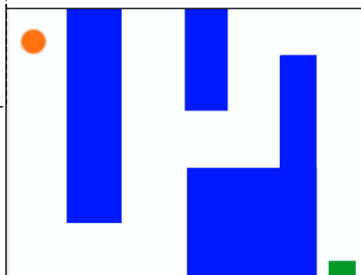
- Programmtext kommentieren
- Programmfehler berichtigen
- Falscheingaben verhindern
- Zusätzliche (gegebene) Bausteine zusammensetzen oder einfügen
- Problemlösung verallgemeinern
(z. B. Eingaben und Variablen statt konstanter Werte)
- Effizienz des Programms erhöhen

Spielideen

DAS LABYRINTH

WIE KANNST DU SCRATCH BENUTZEN, UM EIN INTERAKTIVES SPIEL ZU ERSTELLEN?

In diesem Projekt wirst du ein Spiel erstellen. Dieses Spiel beinhaltet Interaktionen zwischen Sprites, dem Spielstand und Levels. Du bewegst einen Sprite vom Eingang eines Labyrinths zu dessen Ende, ohne die Wände zu berühren.



BEGINNE HIER

- Zeichne einen Labyrinth-ähnlichen Hintergrund und verwende verschiedene Farben für die Wände und für die Endmarkierung des Labyrinths.
- Füge einen Sprite hinzu.
- Mache dein Spiel interaktiv!

DINGE ZUM AUSPROBIEREN

- Füge deinem Spiel mehrere Levels hinzu! Dies kann durch die Verwendung von verschiedenen Bühnenbildern und Sendeblocken durchgeführt werden, die dann den nächsten Level auslösen.
- Verwende den „Variablen“-Block, um den Spielstand im Auge zu behalten!
- Experimentiere mit dem „Stoppuhr“-Block, um deinem Labyrinth neue Herausforderungen hinzuzufügen.



Diese Skripte geben dem Spieler die Kontrolle über die Bewegungen des Sprites im Labyrinth.

Dies sagt deinem Sprite, wo er beginnen soll und markiert den Beginn des Labyrinths.

Dies sagt dem Sprite am Ende des Labyrinths, dass der Spieler gewinnt, wenn der Ball diesen Sprite berührt.

Dies wird dem Sprite von den blauen Wänden abgelesen.

BLÖCKE ZUM SPIELEN



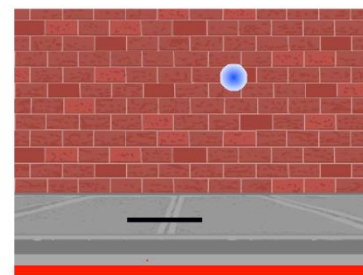
ERLEDIGT?

- Füge dein Projekt dem Spiele-Studio hinzu: <https://scratch.mit.edu/studies/487504>
- Tausche die Spiele mit einem Partner und bespreche eure Kreationen.

PONG

WIE KANNST DU SCRATCH BENUTZEN, UM EIN INTERAKTIVES SPIEL ZU ERSTELLEN?

In diesem Projekt wirst du ein Spiel erstellen. Dieses Spiel beinhaltet Interaktionen zwischen Sprites, dem Spielstand und Levels. Das Spiel ähnelt dem klassischen Spiel „Pong“, bei dem es das Ziel ist, den Sprite davon abzuhalten, an dir vorbei zu kommen.

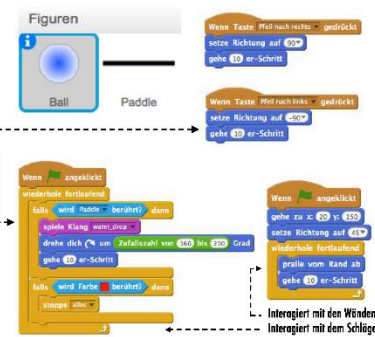


BEGINNE HIER

- Erstelle zwei Sprites: ein Paddle, der vom Benutzer kontrolliert wird und einen Ball, mit dem der Benutzer spielen wird.
- Mache dein Paddle – Sprite interaktiv.
- Bringe dein Spiel zum Leben!

DINGE ZUM AUSPROBIEREN

- Wie kannst du den Schwierigkeitsgrad deines Spiels erhöhen? Das Erstellen von verschiedenen Ebenen, die Verwendung einer Stoppuhr oder den Spielstand im Auge behalten, sind ein paar Beispiele, was man tun könnte.
- Experimentiere damit: Verändere das Aussehen deines Spiels, indem du die Bühnenbilder bearbeitest!
- Verwende verschiedene Tasten, um deinen Sprite zu steuern!



Diese kontrollieren den Ball – wenn das Paddle oder eine Wand berührt wird, bewegt er sich weiter. Wenn Rot berührt wird (was bedeutet, dass sich der Ball an dem Paddle vorbei bewegt), endet das Spiel.

BLÖCKE ZUM SPIELEN



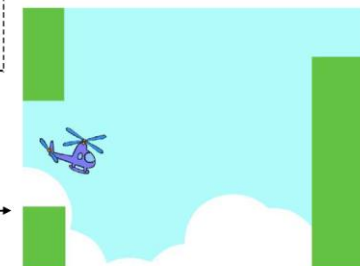
ERLEDIGT?

- Füge dein Projekt dem Spiele-Studio hinzu: <https://scratch.mit.edu/studies/487504>
- Tausche die Spiele mit einem Partner und bespreche eure Kreationen.

KULISSENSPIEL

WIE KANNST DU SCRATCH BENUTZEN, UM EIN INTERAKTIVES SPIEL ZU ERSTELLEN?

In diesem Projekt wirst du ein Spiel erstellen. Dieses Spiel beinhaltet Interaktionen zwischen Sprites, dem Spielstand und Levels. Das Spiel ähnelt Flappy Bird, bei dem es das Ziel ist, ein Objekt daran zu hindern, auf den Boden zu fallen oder andere Objekte zu berühren.

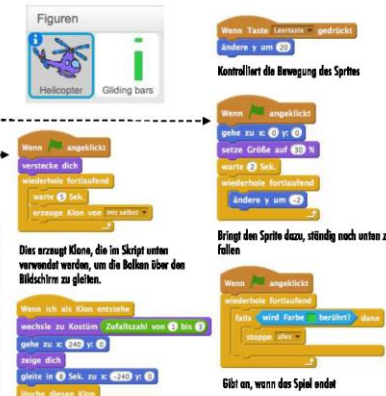


BEGINNE HIER

- Erstelle zwei Sprites: einen, den der Spieler steuern kann (einen Hubschrauber) und einen, den der Spieler meiden soll (gleitende Balken).
- Mache den Hubschrauber interaktiv.
- Bring dein Spiel zum Leben, indem du Skripten hinzufügst, um die gleitenden Balken über die Bühne zu gleiten.

DINGE ZUM AUSPROBIEREN

- Wie kannst du den Schwierigkeitsgrad deines Spiels erhöhen? Das Erstellen von verschiedenen Ebenen, die Verwendung einer Stoppuhr oder den Spielstand im Auge behalten, sind ein paar Beispiele, was man tun könnte.
- Experimentiere damit: Verändere das Aussehen deines Spiels, indem du die Bühnenbilder bearbeitest!
- Verwende verschiedene Tasten, um deinen Sprite zu steuern!



Dies erzeugt Klone, die in Skript unten verwendet werden, um die Balken über den Bildschirm zu gleiten.

Bringt den Sprite dazu, ständig nach unten zu fallen.

Dies sagt dem Sprite, dass das Spiel endet.

BLÖCKE ZUM SPIELEN



ERLEDIGT?

- Füge dein Projekt dem Spiele-Studio hinzu: <https://scratch.mit.edu/studies/487504>
- Tausche die Spiele mit einem Partner und bespreche eure Kreationen.

Anregungen für Programmiererweiterungen

SPIELSTAND

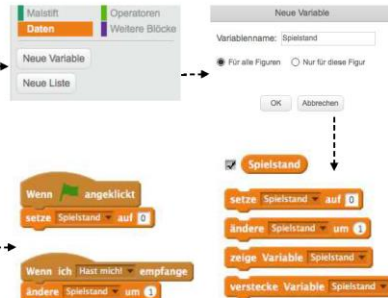
WIE KANNST DU DEN SPIELSTAND IN EINEM SCRATCH-PROJEKT IM AUGE BEHALTEN?

„Fish Chomp“ ist ein Spiel, in dem die Spieler versuchen, so viele Fisch wie möglich zu fangen, indem sie einen Sprite mit der Maus lenken. In dieser Aufgabe wirst du „Fish Chomp“ remixen, indem du einen Spielstand mittels Variablen hinzufügst.



BEGINNE HIER

- ☐ Gehe zur „Fish Chomp“-Projektsite:
<http://scratch.mit.edu/projects/10859244>
- ☐ Klicke auf die Schaltfläche „Neue Variable“ in der Kategorie Daten, um eine Variable für den Spielstand zu erstellen und zu benennen.
- ☐ Experimentiere mit deinem neuen Variablenblock, um einen Spielstand in deinem Projekt zu integrieren!



STECKEN
GEBLIEBEN?

DAS IST OKAY! PROBIER DIESE
DINGE...

ERLEDIGT?

- ☐ Du bist nicht sicher wie du mit Variablen arbeiten sollst? Sieh dir dieses Projekt an, um mehr Informationen zu erhalten:
<http://scratch.mit.edu/projects/2042755>
- ☐ Oder sieh dir dieses Video an: <http://youtu.be/uKq379XkhYw>
- ☐ Erforsche und lerne Codes in Spielen kennen, die einen Spielstand benutzen, um mehr über das Erstellen von Variablen und das Integrieren von Spielständen in ein Projekt zu lernen.

- + Füge dein Projekt dem „Fish Chomp“-Remix-Studio hinzu: <http://scratch.mit.edu/studios/475615>
- + Motiviere dich selbst, mehr zu tun! Wie kannst du den Spielstand nutzen, um den Schwierigkeitsgrad deines Spiels zu erhöhen?
- + Finde ein Spiel, das dich inspiriert und remixe es!

ERWEITERUNGEN

WIE KANNST DU SPIELE IN SCRATCH ERWEITERN UND NEU KONZIPIEREN?

Lass dich für Game-Design begeistern, indem du erweiterte Funktionen zu deinem Scratch-Projekt hinzufügst! Wähle mindestens eine (oder mehrere!) der folgenden Erweiterungen und füge diese deinem zuvor gestarteten Labyrinth, Pong- oder Kulissenspiel hinzu.

BEGINNE HIER

- ☐ Gehe zum „Extensions“-Studio:
<http://scratch.mit.edu/studios/475619>
- ☐ Wähle eine (oder mehrere) der Erweiterungen zum Erforschen.
- ☐ Integriere deine Wahl in dein zuvor gestartetes Spielprojekt!

- + **SPIELSTAND** <http://scratch.mit.edu/projects/1940443>
Demonstriert, wie man einen Spielstand festlegt und verändert. Erhalte jedes Mal zehn Punkte, wenn die Scratch-Katze angeklickt wird.
- + **LEVELS** <http://scratch.mit.edu/projects/1940453>
Demonstriert, wie man Levels verändert. Der Spielstand wird jedes Mal um einen Punkt erhöht, wenn die Leertaste gedrückt wird. Der Level wird pro zehn Punkte um eins erhöht.
- + **STOPPUHR** <http://scratch.mit.edu/projects/1940445>
Demonstriert, wie man die Stoppuhr benutzt. Verwende die Maus, um die Scratch-Katze in Richtung Gabo zu bewegen.
- + **GEGNER** <http://scratch.mit.edu/projects/1940450>
Demonstriert, wie man einen Gegner hinzufügt. Vermeide den Tennisball, indem du die Pfeiltasten „nach oben“ und „nach unten“ benutzt.
- + **BELOHNUNGEN** <http://scratch.mit.edu/projects/1940456>
Demonstriert, wie man Gegenstände sammelt. Verwende die Pfeiltasten, um die Scratch-Katze herumzubewegen, damit sie die gesuchten Gegenstände sammeln kann.
- + **MAUS** <http://scratch.mit.edu/projects/25192659>
Demonstriert, wie man die Maus programmiert, um das Spiel zu kontrollieren. Bewege die Maus, um das Paddel zu bewegen.
- + **NEUSTART** <http://scratch.mit.edu/projects/25192935>
Demonstriert, wie man eine Schaltfläche zum Neustarten des Spiel erstellen kann.
- + **MENÜ** <http://scratch.mit.edu/projects/25192791>
Demonstriert, wie man einen Menü-Bildschirm am Beginn eines Spiels anzeigen kann. Klicke auf START oder RICHTUNGEN am Menü-Bildschirm.
- + **MULTIPLAYER** <http://scratch.mit.edu/projects/25192711>
Demonstriert, wie man einen weiteren Spieler zum Spiel hinzufügt. Spieler 1 benutzt die Pfeiltasten, um Pica durch das Labyrinth zu navigieren und Spieler 2 benutzt die Tasten W, A, S und D, um Momo durch das Labyrinth zu lenken.

DINGE ZUM AUSPROBIEREN

- + Das „Backpack“ kann ein äußerst nützliches Werkzeug während der Programmierung in Scratch sein. Von Codezeilen, über Musikdateien, Sprites und vieles mehr, kann dort gespeichert werden. Probiere es aus, um Erweiterungen in deine Spielprojekte zu integrieren.
- + Als Alternative kannst du Ideen und Codeschnipsel in deinem Planungsprotokoll skizzieren, welches eine weitere großartige Methode für die Planung ist, wie du deine Erweiterungen integrieren kannst.

ERLEDIGT?

- + Füge eine andere Erweiterung zu deinem Labyrinth, Pong oder Kulissenspiel hinzu.
- + Fordere dich heraus, mehr zu tun! Spiele deine Erweiterungen weiter durch und füge sie deinen Spielen hinzu.
- + Hilf einem Nachbarn!
- + Teile dein Projekt mit einem Nachbarn und gebt einander Feedback zu euren Spielen.

<https://eis.ph-noe.ac.at/wp-content/uploads/2017/05/kreative-informatik-shb.pdf>

Anregungen für Programmiererweiterungen

- Spielstände
- Belohnungen
- Zeitmessung
- Gegner
- Mehrere Level
- Spielwiederholung
- Mehrere Spieler
- Bestenlisten
- Spielauswahl (Menüs)
- Videoauswertung

Projektplanung und -entwicklung

<https://eis.ph-noe.ac.at/wp-content/uploads/2017/05/kreative-informatik-shb.pdf>

PROJEKT PLANUNG

PROJEKTPLÄNE VON:

Verwende die Fragen unten, um über die erforderlichen Elemente für die Entwicklung deines Projekts nachzudenken.

MEIN PROJEKT

Beschreibe das Projekt, das du erstellen möchtest.

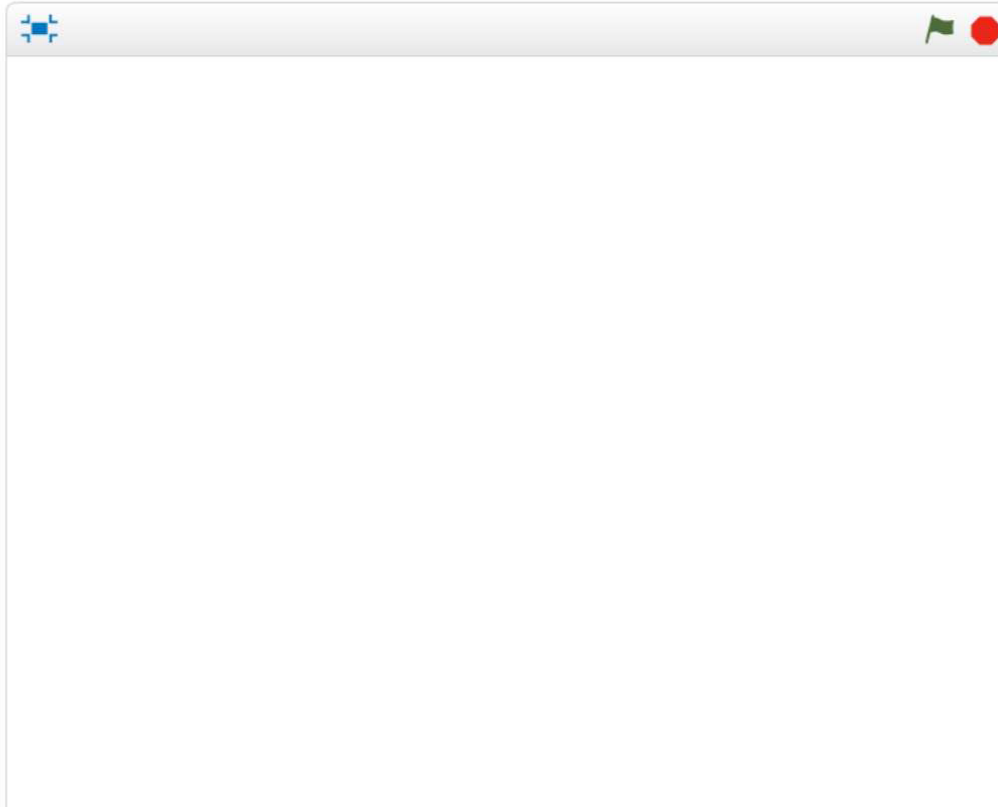
Liste die erforderlichen Schritte auf, um dein Projekt zu erstellen.

MEINE RESSOURCEN

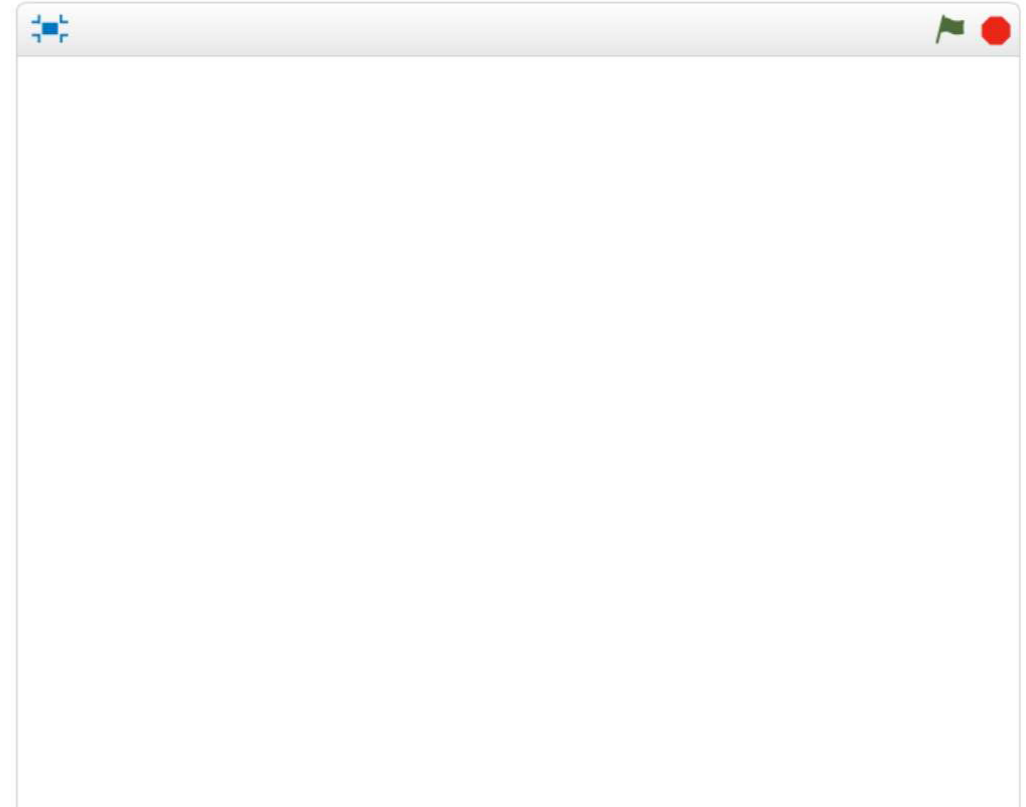
Welche Ressourcen (z. B. Leute, Beispielprojekte) hast du bereits?

Welche Ressourcen (z. B. Leute, Beispielprojekte) könntest du benötigen, um dieses Projekt zu entwickeln?

MEINE PROJEKTSKIZZEN



Was passiert gerade? Was sind die wichtigen Elemente?



Was passiert gerade? Was sind die wichtigen Elemente?

PROJEKT FEEDBACK

FEEDBACK FÜR: _____

PROJEKTTITEL: _____

ROT, GELB, GRÜN

FEEDBACK VON	[ROT] Gibt es etwas, das nicht funktioniert hat oder verbessert werden könnte?	[GELB] Gibt es etwas Verwirrendes oder könnte etwas anders gemacht werden?	[GRÜN] Was hat gut funktioniert oder was hat dir an dem Projekt besonders gut gefallen?

DIESE PROJEKTELEMENTE KÖNNTEN HILFREICH SEIN, WENN MAN SIE BERÜCKSICHTIGT:

- + Klarheit: Hast du verstanden, was das Projekt bewirken soll?
- + Merkmale: Welche Besonderheiten weist dieses Projekt auf? Funktioniert das Projekt wie erwartet?
- + Wirkung: Wie fesselnd ist dieses Projekt? Ist es interaktiv, originell, anspruchsvoll, lustig oder interessant? Wie ist es dir bei der Arbeit ergangen?

PROJEKT UPDATE

PROJEKT-UPDATE VON:

Besprich deinen Design-Fortschritt mit deinem Team und skizziere einen Plan für die nächsten Schritte basierend auf dem erhaltenen Feedback.

PROJEKTFORTSCHRITT

Was war bis jetzt dein Lieblingsteil in diesem Prozess?

Welche Teile deines Projekts müssen noch bearbeitet werden?

NÄCHSTE SCHRITTE

Welche Teile deines Projekts wird jedes Gruppenmitglied als nächstes bearbeiten?

Wobei könntest du Hilfe benötigen, um Fortschritte zu machen?

Dokumentieren und Präsentieren

Dokumentieren

- Kommentieren des Programmtextes
- Bedienungsanleitung (Text, grafisch, Video)
- Werbevideo
- „Making of“-Dokumentation
- „Outtakes“
- Portfolio-Erstellung
- Postererstellung

Präsentieren

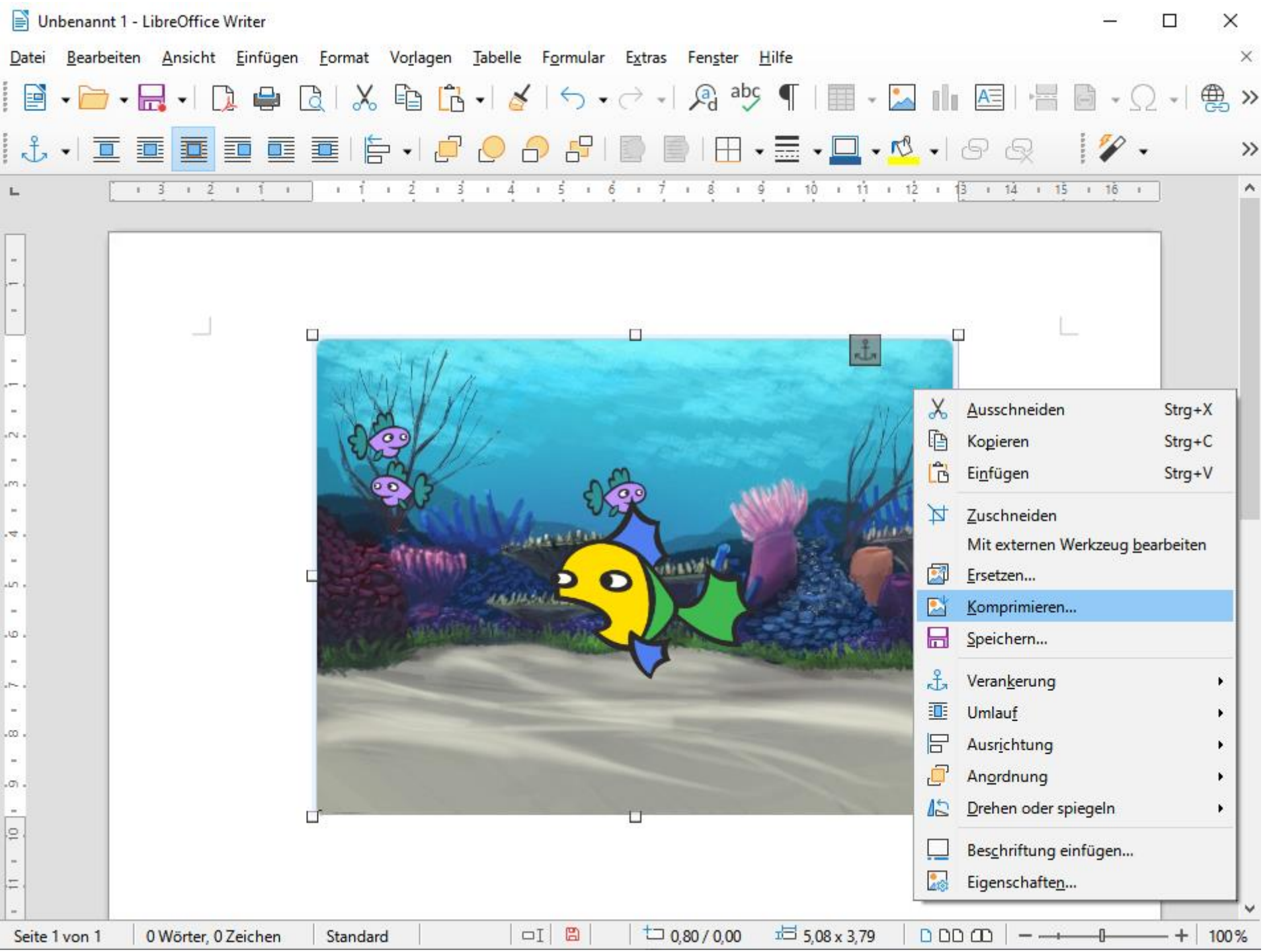
Und dann? Führen **alle allen alles** vor?

Lauf längencodierung

Motivation

Verwendung von Bildern in der Dokumentation → Dateigröße

Probleme beim Versand großer Mediendateien



Weniger mitnehmen oder ordentlich packen?

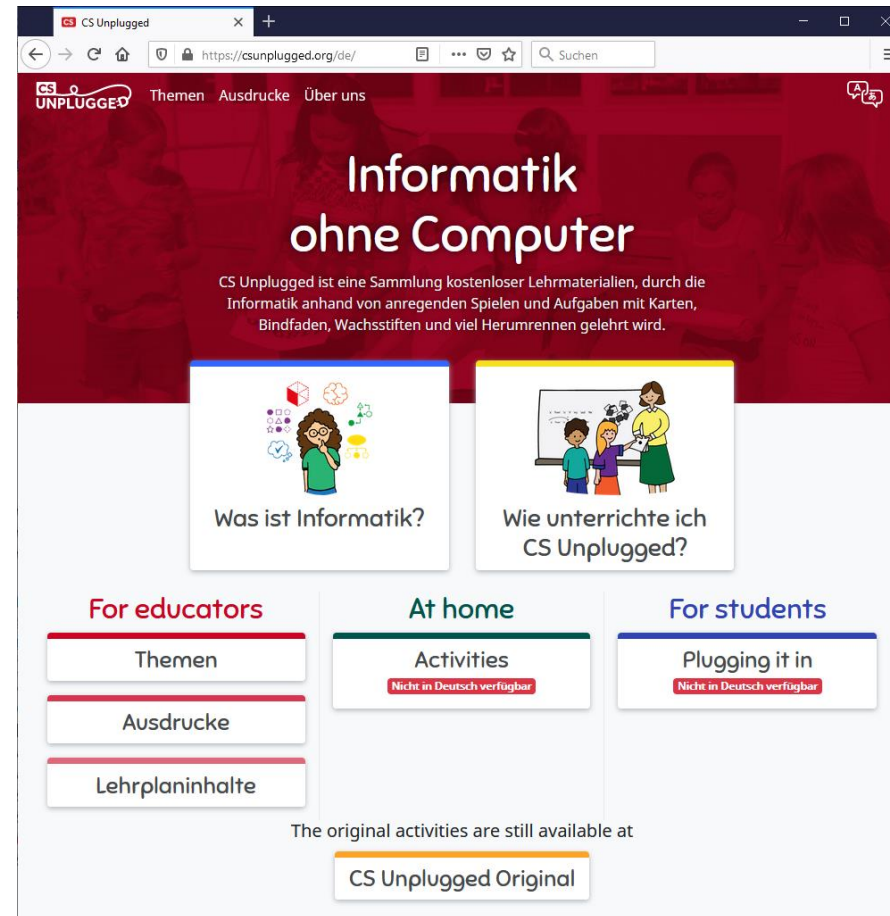


lograstudio@pixabay

Bilder komprimieren



Lauf längenkomprimierung mit CS unplugged



<https://classic.csunplugged.org/image-representation/>

Zeilenkomprimierung

Zeilenkomprimierung

0, 1, 1, 1, 0
0, 0, 0, 0, 1
0, 1, 1, 1, 1
1, 0, 0, 0, 1
1, 0, 0, 0, 1
0, 1, 1, 1, 1

	■	■	■	
				■
	■	■	■	■
■				■
■				■
	■	■	■	■

30

Zeilenkomprimierung

0, 1, 1, 1, 0

0, 0, 0, 0, 1

0, 1, 1, 1, 1

1, 0, 0, 0, 1

1, 0, 0, 0, 1

0, 1, 1, 1, 1

	■	■	■	
				■
	■	■	■	■
■				■
■				■
	■	■	■	■

1, 3, 1

4, 1

1, 4

0, 1, 3, 1

0, 1, 3, 1

1, 4

30

Zeilenkomprimierung

0, 1, 1, 1, 0
 0, 0, 0, 0, 1
 0, 1, 1, 1, 1
 1, 0, 0, 0, 1
 1, 0, 0, 0, 1
 0, 1, 1, 1, 1

1, 3, 1
 4, 1
 1, 4
 0, 1, 3, 1
 0, 1, 3, 1
 1, 4

30

17

Zeilenkomprimierung

0, 1, 1, 1, 0
 0, 0, 0, 0, 1
 0, 1, 1, 1, 1
 1, 0, 0, 0, 1
 1, 0, 0, 0, 1
 0, 1, 1, 1, 1

30

	■	■	■	
				■
	■	■	■	■
■				■
■				■
	■	■	■	■

1, 3, 1
 4, 1
 1, 4
 0, 1, 3, 1
 = 4
 = 3

15

Zeilenkomprimierung

0, 1, 1, 1, 0
 0, 0, 0, 0, 1
 0, 1, 1, 1, 1
 1, 0, 0, 0, 1
 1, 0, 0, 0, 1
 0, 1, 1, 1, 1

30

	■	■	■	
				■
	■	■	■	■
■				■
■				■
	■	■	■	■

1, 3, 1
 4, 1
 1, 4
 0, = 1
 = 4
 = 3

14

Zeilenkomprimierung

0, 1, 1, 1, 0
 0, 0, 0, 0, 1
 0, 1, 1, 1, 1
 1, 0, 0, 0, 1
 1, 0, 0, 0, 1
 0, 1, 1, 1, 1

30

1, 3, ~~1~~
 4, ~~1~~
 1, ~~4~~
 0, = 1
 = 4
 = 3

11

Zeilenkomprimierung

0, 1, 1, 1, 0
0, 0, 0, 0, 1
0, 1, 1, 1, 1
1, 0, 0, 0, 1
1, 0, 0, 0, 1
0, 1, 1, 1, 1

30

	■	■	■	
				■
	■	■	■	■
■				■
■				■
	■	■	■	■

1, 3

4

1

0, = 1

= 4

1

10

Nonogramme (Non Ishida, 1986)

leeres Nonogramm

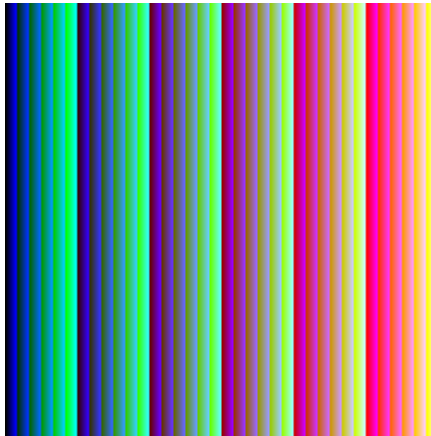
				2	2			
	0	9	9	2	2	4	4	0
0								
4								
6								
22								
22								
6								
4								
2								
2								
2								
0								

gelöstes Nonogramm

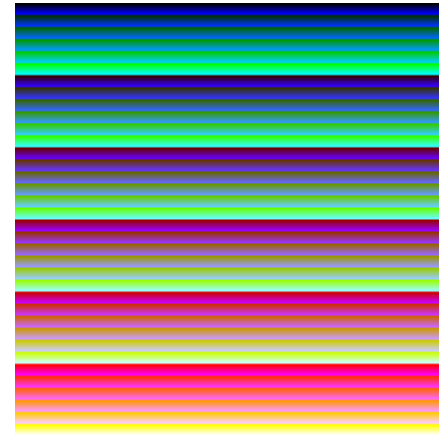
				2	2			
	0	9	9	2	2	4	4	0
0								
4								
6								
22								
22								
6								
4								
2								
2								
2								
0								

Querstreifen machen schlank!

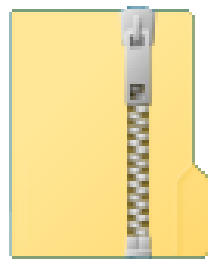
53.698 Bytes



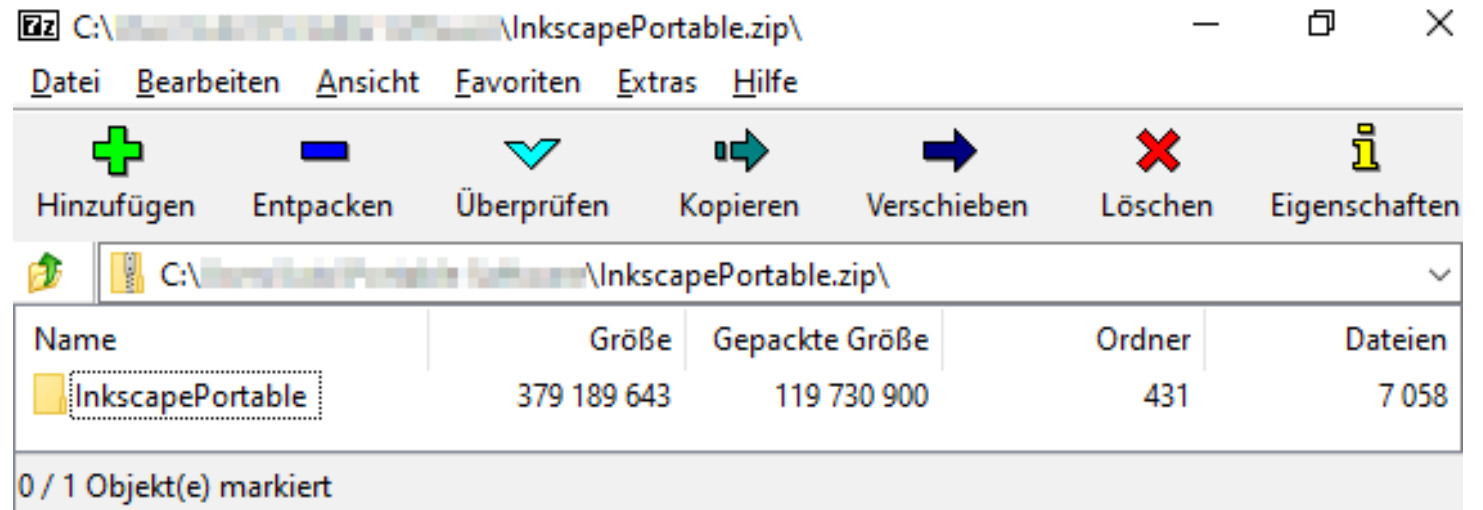
5.978 Bytes



Dateien „zippen“

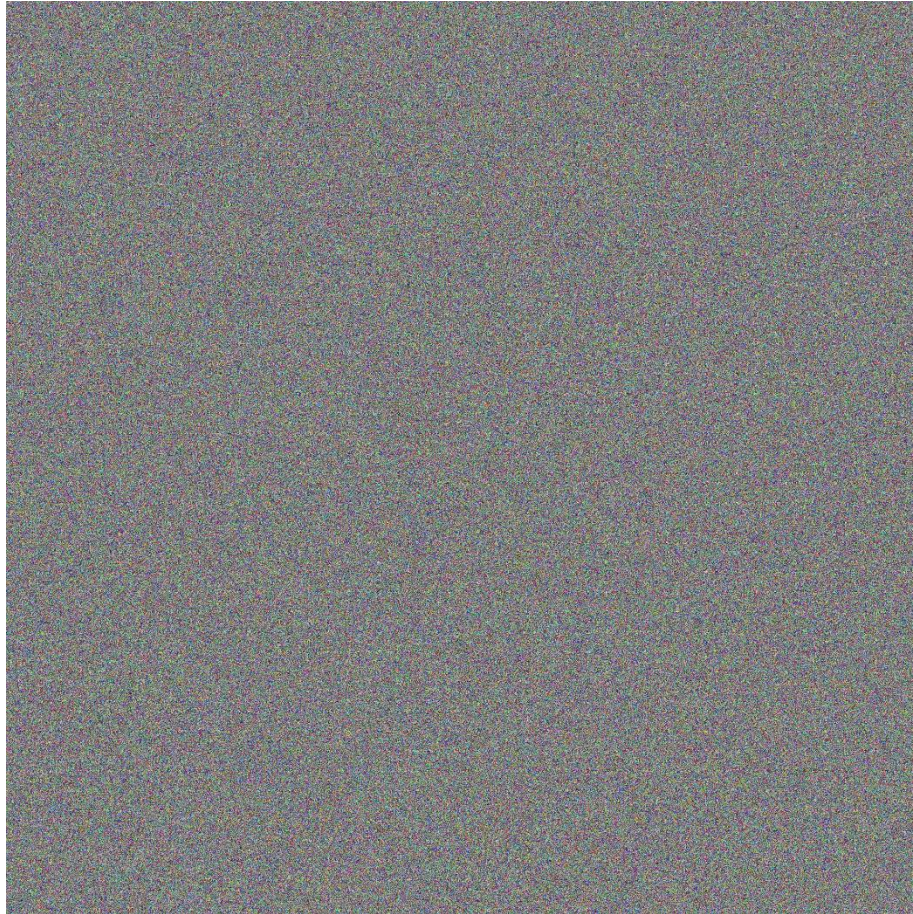


zipdatei.zip



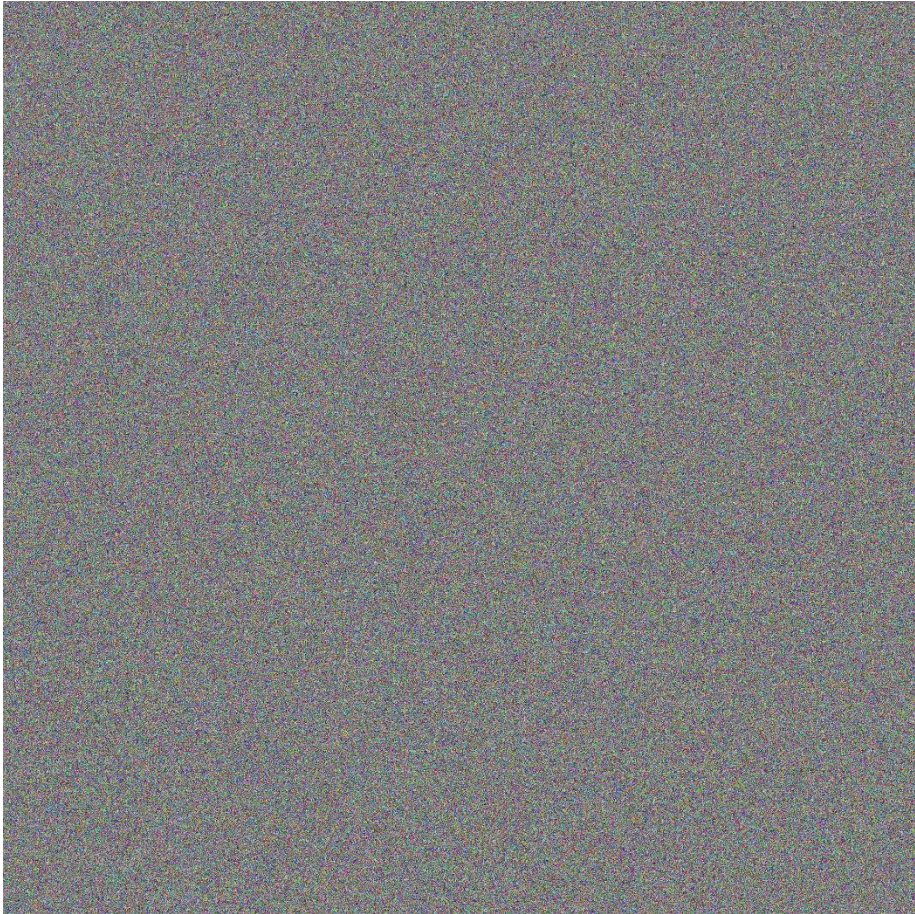
- Zusammenführen mehrerer Dateien zu einer einzigen Datei
- Komprimieren dieser Datei

Schätzfrage



Wie groß ist eine ursprünglich 50.331.702 Bytes große bmp-Bilddatei (unkomprimiert) nach der Umwandlung in eine (komprimierte) png-Datei?

Grenzen der Komprimierung



- Regelmäßigkeiten
 - Redundanzen
- sind Voraussetzung für die Komprimierfähigkeit.

Komprimierung und Verlust

Verlustfrei

- Texte
- Formale Daten
- Numerische Werte
- Masteraufnahmen
- Qualitätsbilder (*.png)

Verlustbehaftet

- „Gebrauchsbilder“ (*.jpg)
- Musik und Ton (*.mp3)
- Videocodecs (*.mp4)

Möglichkeiten der Leistungsbewertung

Praktische Leistungen

- Programmierleistung
- Spielidee und -gestaltung
- Dokumentation

Kurztest

- mit *praktischen* Aufgaben zur Komprimierung
(mit Zettel und Stift sowie am Rechner)