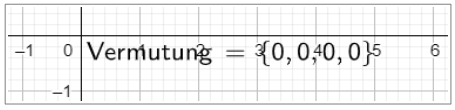
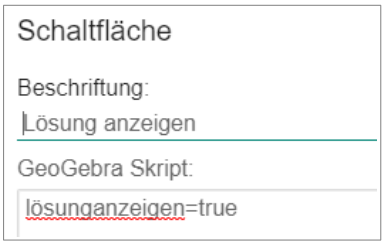
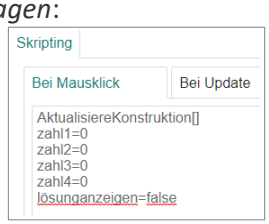
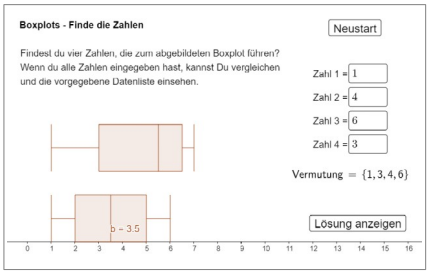
Die Dateien sind als Aktivität unter https://www.geogebra.org/m/ap3ymga2 abrufbar und können auf Tablet oder PC genutzt werden. Mithilfe der Anleitung kann die Erstellung nachvollzogen und variiert werden.
Die Lernumgebung gibt aus vier zunächst unbekannten natürlichen Zahlen einen Boxplot vor, der über die Eingabe von vier Zahlen zu erzeugen ist. Später können die eigene und die vorgegebene Lösung miteinander verglichen werden.

Basisversion (mit Rückmeldung nach Lösungsversuch)

Symbol	Inhalt / Beschreibung	Alternativ Eingabe / Bemerkungen
Eingabezeile	Zunächst werden vier Zahlvariablen *zahl1=0* bis *zahl4=0 definiert*; diese benötigen wir später für die Werte der Eingabefelder. *Ab classsic 6 werden Zahlvariablen als Schieberegler angelegt, die Parameter muss man anschließend anpassen oder wie rechts direkt übergeben.*	*zahl1=schieberegler(0,10,1)* *zahl2=schieberegler(0,10,1)* *zahl3=schieberegler(0,10,1)* *zahl4=schieberegler(0,10,1)*
	Im Grafikfenster werden vier Eingabefelder mit der Beschriftung „Zahl 1 =“, „Zahl 2 =“ o.Ä. angelegt und mit der jeweiligen Variable zahl1, zahl2, u.s.w. verknüpft: 	In diese Eingabefelder geben SchülerInnen die Zahlenwerte ein, die zu dem Boxplot führen sollen, der dem vorgegebenen Boxplot gleicht. Varianten der Eingabe über Tabellen sind möglich, an dieser Stelle aber unnötig kompliziert. Alternativ: Mit dem Befehl *Eingabefeld(zahl1)* kann man direkt in der Eingabezeile definieren, muss dann allerdings nachträglich im Eigenschaftsmenu die Beschriftung hinzufügen.
Eingabezeile	Die vier eingegebenen Zahlen werden zu einer Liste mit dem Namen *zahlen* zusammengefasst.	*zahlen = {zahl1, zahl2, zahl3, zahl4}*
Eingabezeile	*Vermutung=(Sortiere(Folge(Element(zahlen,i),i,1,4)))* erzeugt den aktuellen Lösungsvorschlag als geordnete Liste auf Grundlage der vier einzugebenden Zahlen. *Formeltext(Vermutung, true,true)* legt die Liste als Objekt *Text1* an, das im Ursprung des KS angezeigt wird:  Das Textobjekt muss noch passend verschoben werden.	Befehle siehe links, Erläuterung: Mit *Element[zahlen,i]* greift man auf das i-te Elemente der Liste *zahlen* zu, *Folge* ermöglicht den Zugriff auf die vier Elemente nacheinander, *Sortiere* ist selbsterklärend und *Formeltext* erlaubt schließlich die Ausgabe des Objekts (hier die Liste) als Text im Grafikfenster (die beiden Boolschen Variablen steuern dabei die Anzeige des Objektnamens und der aktuellen Werte).
Eingabezeile	*Lösung=Sortiere(Folge(Zufallszahl(1,10),i,1,4))* erzeugt die Zufallsliste *Lösung* mit 4 Zahlen, die der Größe nach aufsteigend sortiert wird.	Befehl siehe links
Eingabezeile	**Lösungsliste** als Formeltextobjekt *Text2* anlegen und einblenden (danach noch passend verschieben).	*FormelText(Lösung, true, true)*

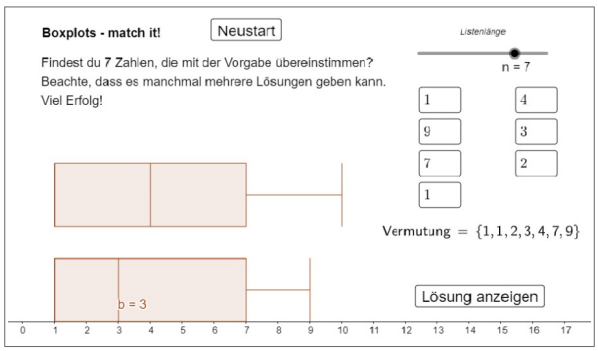
Eingabe-zeile	**Sichtbarkeit der Lösung (Text2)** Die Boolsche Variable *lösunganzeigen* wird definiert und auf den Wert false gesetzt. Im Eigenschaftsmenu des Objektes *Text2* wird (unter dem Reiter *Erweitert)* als Bedingung für dessen Sichtbarkeit der Variablenname *lösunganzeigen* eingetragen.	*lösunganzeigen=false* Diese Variable wird auf *true* gesetzt, wenn später die Schaltfläche *Lösung anzeigen* (s.u.) angeklickt wird.
OK	**Schaltfläche1 „Lösung anzeigen“** Nach Wahl des grafischen Werkzeuges klickt man an die gewünschte Stelle und gibt in der Abfragemaske die Beschriftung *Lösung anzeigen* und beim Skript den Befehl *lösunganzeigen=true* ein:  Hinweis: Schaltflächen wie hier *Schaltfläche1* werden i.d.R. als Hilfsobjekte angelegt und erscheinen daher zunächst nicht in der Algebraansicht.	Legt man die Schaltfläche über die Eingabezeile ein, dann muss man noch im Eigenschaftsmenu der Schaltfläche unter dem Reiter *Skripting → Bei Mausklick* den Skriptbefehl nachtragen. Falls die Kurzform des Skriptbefehls *(lösungsanzeigen=true)* Probleme bereiten sollte, schreibt man ihn aus: *SetzeWert(lösunganzeigen,true)* *Hilfsobjekte einblenden!* 
Eingabe-zeile	**Sichtbarkeit der Schaltfläche1 einschränken** Die Boolesche Variable *alle4eingegeben* wird wie rechts angegeben definiert und anschließend im Eigenschaftsmenu der *Schaltfläche1* als Bedingung für deren Sichtbarkeit eingetragen (unter dem Reiter *Erweitert*). Nur wenn alle vier Eingabefelder ausgefüllt wurden, ist Schaltfläche1 somit sichtbar.	*alle4eingegeben=Wenn(zahl1 ≠ 0 ∧ zahl2 ≠ 0 ∧ zahl3 ≠ 0 ∧ zahl4 ≠ 0, true, false)* Wichtige GeoGebra-Kurzschreibweisen != erzeugt ≠ == erzeugt $\stackrel{?}{=}$ && erzeugt ∧ || erzeugt ∨
Eingabe-zeile	**Boxplots** definieren und anzeigen lassen Syntax: *boxplot(y-Abstand,y-Skalierung, Datenliste)* *(Hinweis: „y-Skalierung“ = halbe Höhe des Boxplots)*	*Boxplot(4, 1, Lösung)* *Boxplot(1,1,Vermutung)*
OK	**Schaltfläche2 „Neustart“ einrichten** als *Skript* (*Bei Mausklick) eintragen*: *AktualisiereKonstruktion[]* *zahl1=0* *zahl2=0* *zahl3=0* *zahl4=0* *lösunganzeigen=false* 	

Damit sollte die Basisversion funktionsbereit sein! Ein Aufgabentext kann ergänzt und das Layout angepasst werden (vgl. Bild rechts z.B. ohne Koordinatengitter und y-Achse).

Erweiterung1: Wahl der Listenlänge (zur Differenzierung)

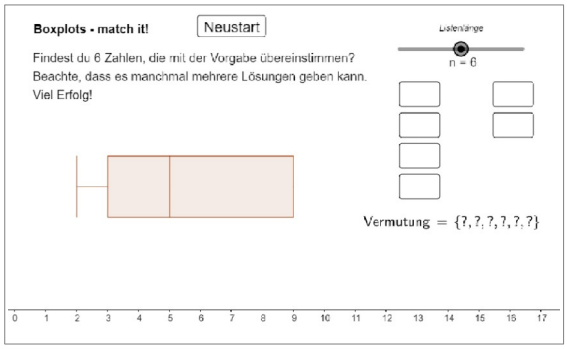
Die Lernumgebung basiert auf der zuvor erstellten Basisversion und soll nun auf acht Zahlen erweitert werden, wobei die SuS zur Differenzierung selbst die Listenlänge über einen Schieberegler wählen können.

Inhalt / Beschreibung	Eingabezeile / Hinweise:
Der Schieberegler wird zur Steuerung der Listenlänge benötigt.	*n=Schieberegler(4,8,1)*
Neue Zahlvariablen *Eine gute Gelegenheit, um sich mit den Shortcuts „Strg“+“c“ (Kopieren) und „Strg“+“v“ (einfügen) vertraut zu machen, copy & paste!*	*zahl5=schieberegler(0,10,1)* *zahl6=schieberegler(0,10,1)* *zahl7=schieberegler(0,10,1)* *zahl8=schieberegler(0,10,1)*
Neue Eingabefelder (*copy & paste!)* *Die Länge aller Eingabefelder sollte man einheitlich anpassen* *Tipp: Alle 8 eingabefelder auswählen und nur einmal anpassen:* im Eigenschaftsmenu unter dem Reiter *Darstellung (z.B. „4“)*	*Eingabefeld5=Eingabefeld(zahl5)* *Eingabefeld6=Eingabefeld(zahl6)* *Eingabefeld7=Eingabefeld(zahl7)* *Eingabefeld8=Eingabefeld(zahl8)*
Sichtbarkeit der neuen Eingabefelder Im Eigenschaftsmenu der neuen Eingabefelder gibt man als Bedingung für die Sichtbarkeit n>=5, n>=6, u.s.w. ein, damit bei Listenlänge n auch nur n Eingabefelder sichtbar sind.	Tipp: Eigenschaftsfenster geöffnet lassen und die Objekt nacheinander anklicken, die Eingabe „n>=…“ jedes Mal mit <enter> abschließen.
Listen umdefinieren (erweitern) Bei den Listen *Vermutung* und *Lösung* muss die Definition (nach Doppelklick oder im Eigenschaftsmenu) angepasst werden, da die Laufvariable i nun von 1 bis n (statt 1 bis 4) laufen soll. Dazu genügt es, die „4“ am Ende durch „n“ zu ersetzen. :) Die Liste *zahlen* wird um die vier neuen Zahlvariablen ergänzt.	*Vermutung=Sortiere(Folge(Element(zahlen,i),i,1,**n**))* *Lösung=Sortiere(Folge(Zufallszahl(1 , 10), i, 1, **n**))* *zahlen={zahl1, zahl2, zahl3, zahl4, zahl5, zahl6, zahl7, zahl8}*
Sichtbarkeit der Schaltfläche „Lösung anzeigen“ Hier wurde eine „Update“-Skriptlösung gewählt. Das Skript prüft ob alle n Felder ausgefüllt sind und wird bei jeder Aktion (jedem Update) im Hintergrund ausgelöst. Es kann daher prinzipiell bei fast jedem Objekt angedockt werden. Hier wurde der Boxplot b gewählt. Für die Prüfung benötigt man zwei Variablen, die zuvor wie rechts definiert werden sollten. Dann kann man im Eigenschaftsmenu des Boxplots unter dem Reiter Skripting (bei Update) folgendes Prüf-Skript eintragen: *m=0* *Lschaltersichtbar=false* *lösunganzeigen=false* *Hilfsliste=Folge(Wenn(Element(zahlen,i)>0,m=m+1)),i,1,n) * Wenn(m==n,SetzeWert(LSchaltersichtbar,true))* * Ohne Benennung der Folge würde bei jedem Zählvorgang des Prüfskripts eine neue Liste generiert! Da man hier einen Namen vorgibt, wird diese „Hilfsliste“ einmal angelegt und fortlaufend überschrieben.	Etwas aufwändiger, da die Sichtbarkeit in der Basisversion nur mithilfe der *alle4eingeben* geregelt werden konnte, was jetzt nicht mehr funktioniert (Warum?). Zählvariable m für die Anzahl der von „0“ abweichenden Einträge: *m=schieberegler(0,8,1)* Boolsche Variable für die Sichtbarkeit der Schaltfläche: *Lschaltersichtbar=false* Im Skript werden die Einträge gezählt und deren Anzahl m mit der Listenlänge n verglichen. Nur wenn alle n Zahlen eingegeben wurden, wird die Schaltfläche zum Anzeigen der Lösung sichtbar, da *Lschaltersichtbar* auf true gesetzt wird.
Damit es funktioniert , muss man bei der Schaltfläche „Lösung anzeigen“ (unter *Erweitert*) noch die Bedingung der Sichtbarkeit auf *Lschaltersichtbar* aktualisieren. Gute Nerven bei den Tests und der Fehlersuche :) !	

Inhalt / Beschreibung	Eingabezeile / Hinweise:
Aufräumen und Anpassen Die Boolsche Variable *alle4eingeben* aus der Basisversion wird in der Erweiterung nun nicht mehr benötigt und kann gelöscht werden. Beim abschließenden Layout wird man noch einige Objekte sperren oder am Bilschirm anheften. Im Vorschlag rechts wurde der Schieberegler noch mit einem Textfeld beschriftet und die Beschriftungen der Eingabefelder ausgeblendet. Im Aufgabentext wurde die Listenlänge *n* dynamisch eingebunden.	 Screenshot von Erweiterung1 Tipp: Am Ende sollte man noch die Algebraansicht kontrollieren, um generierte Fehlobjekte zu löschen und vor dem Speichern „aufzuräumen“.

Erweiterung 2: Schwierigkeit steigern – Boxplot während Eingabe ausblenden

Zunächst war es erwünscht und hilfreich, dass der Boxplot unten fortlaufend sichtbar bleibt und dynamisch aktualisiert wird. Die SuS sollten die Datenliste allerdings auch ohne diese unterstützende Visualisierung eingeben können. Dies lässt sich über den Reset der Zahlenwerte beim Neustart einfach realisieren:

Inhalt / Beschreibung	Eingabezeile / Hinweise:
Skript der Schaltfläche2 „Neustart“ anpassen: AktualisiereKonstruktion[] zahl1=**?** … zahl8=**?** lösunganzeigen=false Hinweis: Setzt man beim Neustart die Zahlenwerte mit dem ? auf unbekannt statt wie vorher auf 0, so kann der Boxplot erst generiert und angezeigt werden, wenn alle n Zahlen eingegeben und damit bekannt sind.	 Screenshot: Erweiterung 2 mit leeren Eingabefeldern

Alternativ könnte man auch die Sichtbarkeit des Boxplots über eine Boolsche Variable steuern und ihn erst einblenden, wenn alle Felder ausgefüllt sind. Hierzu kann die bereits definierte Variable *Lschaltersichtbar* genutzt werden. Die oben skizzierte Variante hat aber den Vorteil, dass das Eingabefeld direkt nach dem Anklicken ausgefüllt werden kann.