

DOKUMENTATION

Von Emran



Inhaltsverzeichnis

Deutsch.....	4
Figuren.....	4
Figur Giga.....	4
Figur Winki.....	4
Definitionsblöcke.....	4
„fliege zur lampe“.....	4
„Freudensprung“.....	4
Wenn die Figur angeklickt wird – Block.....	5
Wenn ich „gefangen“ empfange - Block.....	5
Wenn ich „Schloss offen“ empfange - Block.....	5
Kostüme.....	6
Figur Potion.....	7
Figur Button3.....	7
Empfangsblöcke.....	7
„spiel erklären“.....	7
„spiel ende“.....	7
Kostüme.....	8
Figur Button2.....	8
Empfangsblöcke.....	8
„spiel erklären“.....	8
„spiel ende“.....	8
Kostüme.....	8
Figur Magic Wand.....	9
Funktion.....	9
Startblock.....	9
Wenn „spiel erklären“ empfangen wird – Block.....	9
Wenn „spiel ende“ empfangen wird – Block.....	9
Beim Anklicken der Figur – Block.....	9
Beim drücken der Leertaste - Block.....	10
Kostüme.....	10

Figur Button1.....	11
Funktion.....	11
Definitionsblöcke.....	11
„random xy [speed]“.....	11
Startblock.....	11
Wenn „s“ gedrückt wird – Block.....	12
Wenn „spiel erklären“ empfangen wird.....	12
Wenn die Figur angeklickt wird - Block.....	13
Wenn „spiel start“ empfangen wird - Block.....	13
I. der Spielmechanismus.....	13
II. die Highscore-Berechnung und die Bestimmung von Sieger/Verlierer.....	14
III. Die Frage um nochmal zu spielen.....	14
Figur TXT.....	16
Funktion.....	16
Startblock.....	16
Definition.....	16
„Schrink“.....	16
Empfangsblöcke.....	16
Countdown.....	16
Kostüme.....	17
Figur Wizard.....	18
Funktion.....	18
Wenn das Projekt startet – Block.....	18
Wenn „Zauberer“ empfangen wird – Block.....	18
Wenn „Schloss offen“ empfangen wird – Block.....	18
Wenn „r“ gedrückt wird – Block.....	18
Figur Schloss.....	20
Funktion.....	20
Startblock.....	20
Wenn „Rätsel“ empfangen wird – Block.....	20
Wenn „gefangen“ empfangen wird (I) – Block.....	21
Wenn die Figur angeklickt wird – Block.....	21

Wenn „Animation: Schloss schütteln“ empfangen wird – Block.....	21
Wenn „gefangen“ empfangen wird (II) – Block.....	22
Definitionsblöcke.....	22
„stellen berechner“.....	22
„Code rechner && Rätsel ersteller“.....	23
„Ziffer 1 berechnen“.....	23
„Ziffer 2 berechnen“.....	24
„Ziffer 3 berechnen“.....	24
„Ziffer 4 berechnen“.....	25
„Ziffer 5-10 berechnen“.....	25
„Animation: Schloss schütteln“.....	26
Kostüme.....	27
Figur Rätsel.....	27
Funktion.....	27
Definitionsblock.....	27
„random xy {speed}“.....	27
Startblock.....	27
Wenn ich als Klon entstehe – Block.....	28
Kostüme.....	28
Bühnenbild.....	29
Scripte.....	29
Startblock.....	29
Bühnenbilder.....	29
DALL·E 2024-11-12 12.....	29
Moon.....	30
Woods.....	30
Variablen.....	31
Listen.....	32
Zufallsverteilung.....	34

Deutsch

Figuren



Figur Giga

Keine Modifizierung vorhanden!

Figur Winki

Definitionsblöcke

„*fliege zur lampe*“

Dieser Block ist teil Aufgabe 2.

Winki dreht sich um 180 Grad und gleitet zur anderen Laterne.



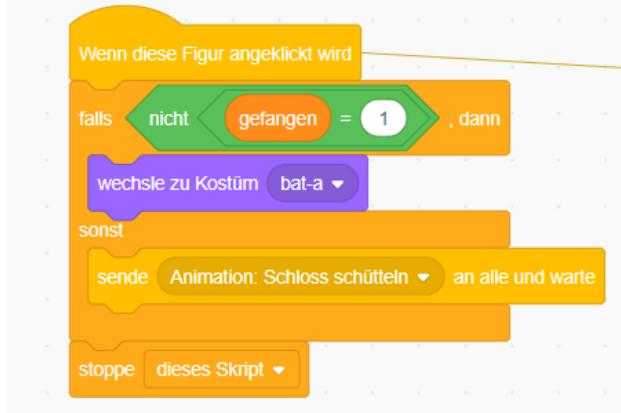
„*Freudensprung*“

Dieser Block ist teil Aufgabe 3.

Erst wechselt Winki zum Kostüm b und gleitet nach oben dann wechselt es zu Kostüm c und gleitet dann nach unten. Schließlich ändert Winki das Kostüm zu a.



Wenn die Figur angeklickt wird – Block



Wenn die Figur angeklickt wird überprüft es, ob Winki **nicht** gefangen ist.

Wenn dies zutrifft, dann wechselt es zum Kostüm **bat-a**.

Wenn dies nicht zutrifft, dann sendet Winki „**Animation: Schloss schütteln**“ an alle und wartet.

Am Ende beider Falls-Abzweigungen wird dieses Script gestoppt. Das Falls-Block dient um den Benutzer davon abzuhalten, während der Zauberer Szene Winki durch den Klick der Figur, u. a. den Kostüm zu wechseln. Dies wird benötigt, weil während der Szene der Kostüm zum „gefangenem Bre“ geändert und gesperrt wird.

Wenn ich „gefangen“ empfange - Block



Sobald gefangen empfangen wird, setzt das Script → [**gefangen** = **1**];

Dann wechselt Winki zum Kostüm „gefängener bre“ und Winki wird um 30 [Einheiten] größer, mithilfe des „wiederhole 30 mal → ändere Größe um 1“- Block.

Dieser Block verschafft auch zu dem eine Animation die die Figur größer wirken lässt.

Wenn ich „Schloss offen“ empfange - Block

Sobald das Rätsel gelöst wurde, wird die Variable [**gefangen**] auf „0“ gesetzt.

Dann wechselt Winki in Kostüm „bat-c“ und Winki wird mit Verzögerung um 30 [Einheiten in der Größe] verkleinert.

Dann wechselt Winki in Kostüm „bat-a“.

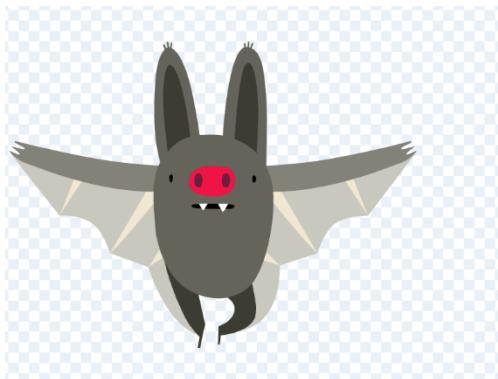
Dies ist eine Animation, die zeigen soll, dass die Fledermaus am Ende frei ist.

Dann sagt Winki „Danke für deine Hilfe !!1“ und wartet 2 Sekunden und sagt dann „Tschüss!“

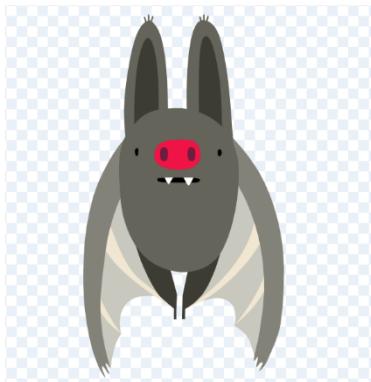
Dann wird erst 5 Sekunden gewartet, damit andere Prozesse zu Ende kommen und dann wird alles beendet.



Kostüme



Kostüm A



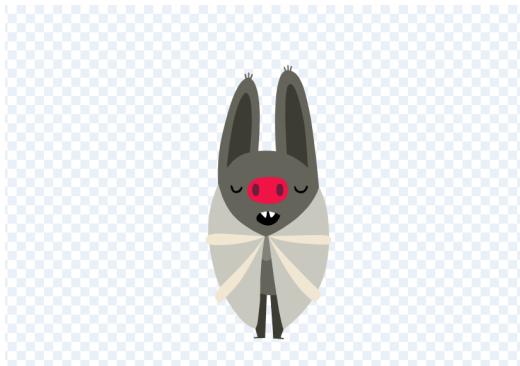
Kostüm B



Kostüm C



Kostüm „gefugener bre“



Kostüm D

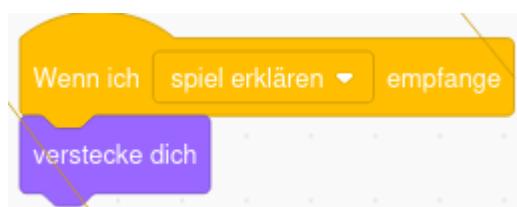
Figur Potion

Keine Modifikation vorhanden!

Figur Button3

Empfangsblöcke

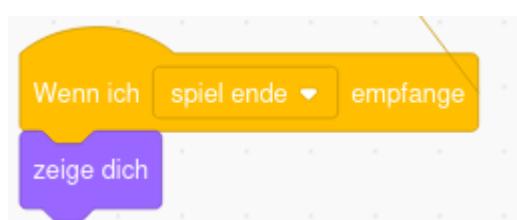
„spiel erklären“



Wenn der Befehl *spiel erklären* empfangen wird, dann versteckt sich der Knopf.

Ich habe dies hin gefügt, damit die Sicht während des Spieles frei bleibt.

„spiel ende“



Wenn das Spiel zu Ende ist, dann zeigt es sich wieder.



Kostüme

Kostüm A

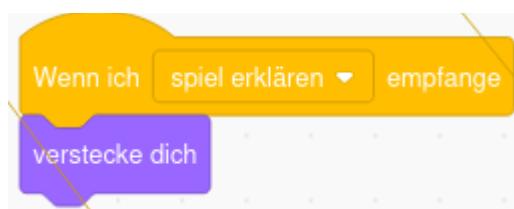


Kostüm B

Figur Button2

Empfangsblöcke

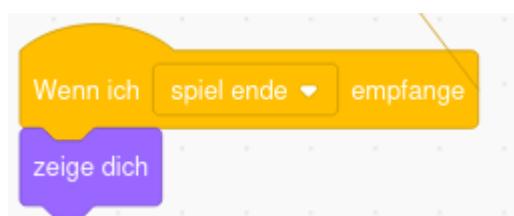
„spiel erklären“



Wenn der Befehl *spiel erklären* empfangen wird, dann versteckt sich der Knopf.

Ich habe dies hin gefügt, damit die Sicht während des Spieles frei bleibt.

„spiel ende“



Wenn das Spiel zu Ende ist, dann zeigt es sich wieder.

Kostüme



Kostüm A

Figur Magic Wand

Funktion

Sie dient als Lösung für Aufgabe 4 und 5, denn sie existiert (siehe Aufgabe 4) und bringt Giga und Winki an einen geheimen Ort (siehe Aufgabe 5).

Startblock



Wenn das Programm startet, dann wird klick auf 0 gesetzt, die Figur versteckt sich und sie geht in der Nähe von Giga auf $[x;y] = [127;-119]$.



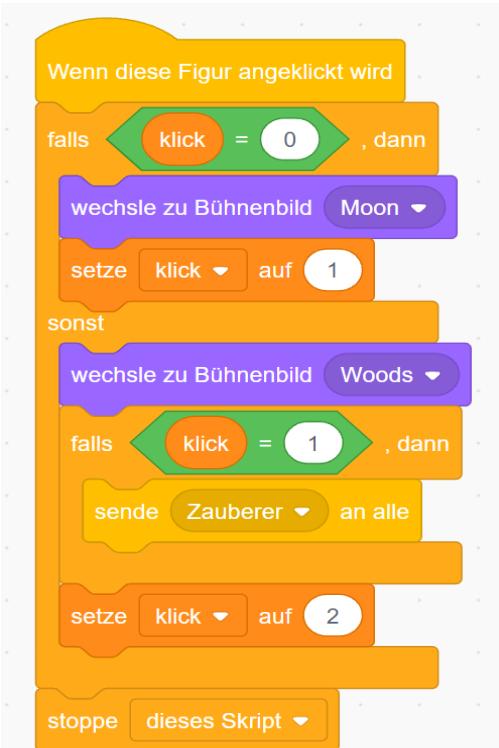
Wenn „spiel erklären“ empfangen wird – Block
Wenn „spiel erklären“ empfangen wird, dann versteckt sich die Figur.



Wenn „spiel ende“ empfangen wird – Block

Wenn „spiel ende“ empfangen wird, dann zeigt sich die Figur.

Beim Anklicken der Figur – Block

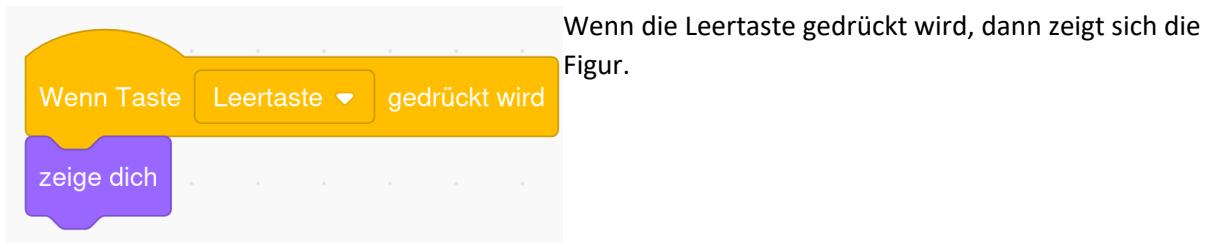


Wenn die Figur angeklickt wird, dann checkt der Script ob die Variable klick gleich 0 ist, wenn dies zutrifft, dann wechselt die Figur zu Bühnenbild „Moon“ und klick wird auf 1 gesetzt.

Wenn dies nicht zutrifft, dann wechselt es das Bühnenbild zu Woods und checkt ob klick gleich 1 ist. Trifft das zu, dann wird Zauberer an alle gesendet. Dies dient dazu, damit Zauberer nicht mehrere Male gesendet wird.

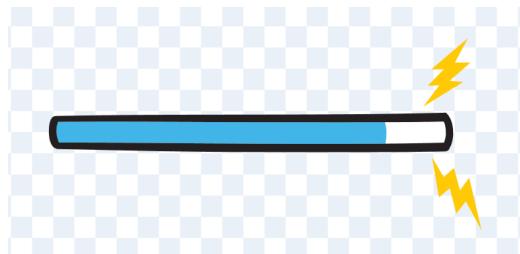
Danach wird klick auf 2 gesetzt und anschließend wird dieses Script gestoppt.

Beim drücken der Leertaste - Block



Wenn die Leertaste gedrückt wird, dann zeigt sich die Figur.

Kostüme



Kostüm 1

Figur Button1

Funktion

Diese Figur ist die Lösung für die Aufgabe 6, denn da muss man ein Klick-spiel, welches für 15 Sekunden andauert, erstellen. Diese Figur beinhaltet die Spielfigur, den Spielmechanismus und den Start/Stopp -mechanismus.



Definitionsblöcke

„random xy [speed]“

Der Definitionsblock random xy setzt die Größe der Figur aus einer Zufallszahl im Intervall [50;80], setzt die y-Koordinate aus einer Zufallszahl im Intervall [-156;106] und die x-Koordinate aus einer Zufallszahl im Intervall [-219;207].

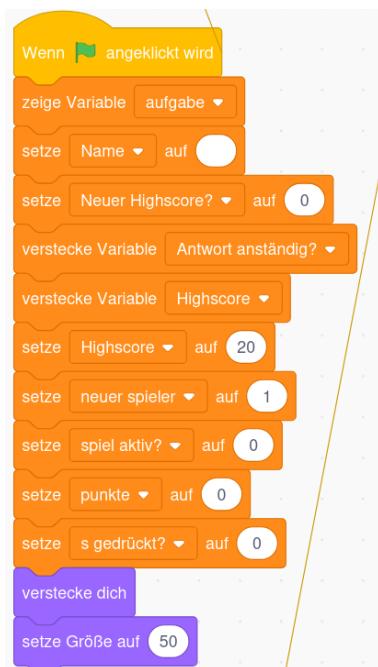
Dann zeigt sich die Figur, wenn sie davor versteckt war.

Und sie wartet „speed“ Sekunden.

„speed“ ist in diesem Block ein Platzhalter für ein Wert. Dieser Wert bestimmt die Geschwindigkeit. Je kleiner der Wert, desto schneller bewegt sich die Figur zu einer

Zufallsposition und Zufallsgröße.

Startblock

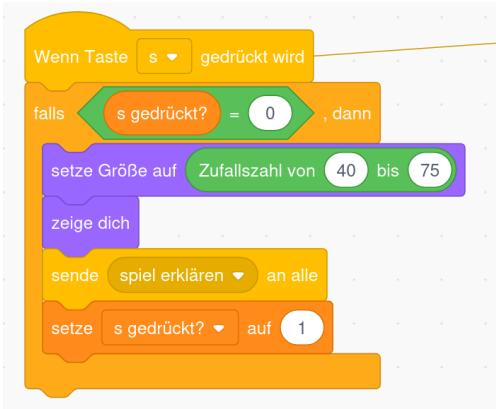


Wenn das Programm startet, dann stellt das hier alle Variablen auf den Startwert.

Die Variable „neuer spieler“ wird auf 1 gestellt, weil das Programm startet und der Highscore wird auf 20 [Sekunden] gestellt und versteckt.

Dann versteckt sich die Figur selber.

Wenn „s“ gedrückt wird – Block

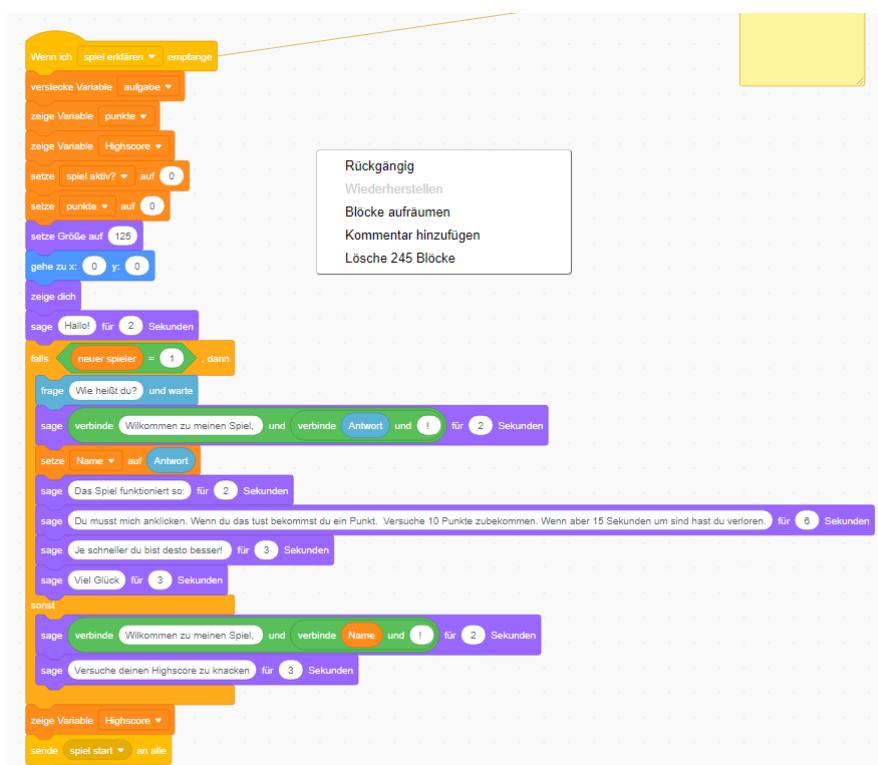


Sobald „s“ gedrückt wird, wird die Figur auf eine Zufallsgröße im Intervall [40;75] gesetzt und sie zeigt sich.

Dann wird „spiel erklären“ an den anderen Figuren übermittelt.

Die Variable „s gedrückt?“ ist da, damit man nur ein einziges mal den Script ausführen kann

Wenn „spiel erklären“ empfangen wird



Sobald „spiel erklären“ empfangen wird, setzt die Figur spezifische Variablen zurück. Dann wird die Größe auf 125 gesetzt und die Figur geht zu x;y: 0;0.

Die Figur zeigt sich und sagt für 2 Sekunden „Hallo“

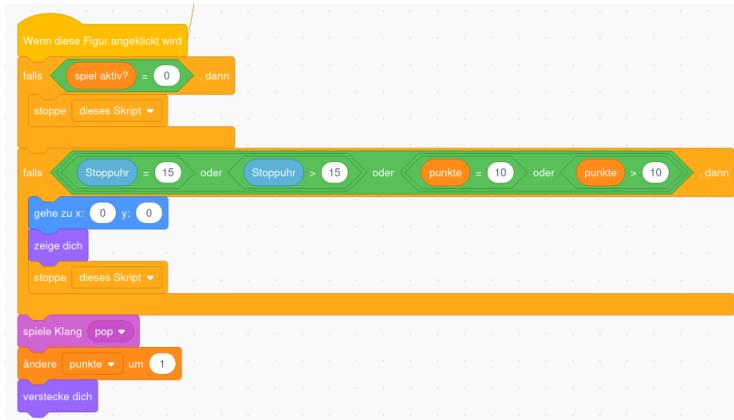
Wenn es sich um einen neuen Spieler handelt, wird der Name gefragt, welches dann in der Variable „name“ gespeichert wird. Dann erklärt die Figur das Spiel.

Wenn es ein Spieler ist, der nochmal spielen will, wird sein Name gesagt und dann sagt die Figur, dass der Spieler seinen Highscore knacken

solle.

Nach dem „Falls/Sonst“-Block wird die Variable „Highscore“ gezeigt und es wird „spiel start“ an alle Figuren übermittelt.

Wenn die Figur angeklickt wird - Block



Wenn die Figur angeklickt wird, dann checkt es ob es ein aktives Spiel gibt. Wenn dies nicht zutrifft, dann wird dieses Script gestoppt. (Dies dient, damit man nicht mogeln kann.)

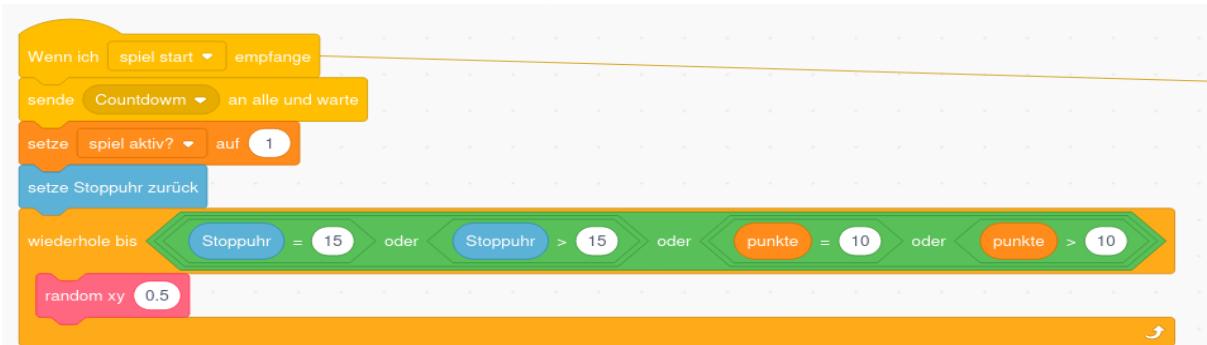
Dann wird gecheckt ob das Spiel nicht zu Ende ist. Wenn das zutrifft dann geht die Figur auf 0;0, zeigt sich und der Script wird gestoppt.

Wenn dies nicht zutrifft, dann wird der Pop Sound gespielt, die Variable „punkte“ um eins erhöht und die Figur versteckt sich.

Wenn „spiel start“ empfangen wird - Block

Dieser Block ist in drei Teile unterteilt. Der Block läuft von **I.** zu **II.** und endet mit **III.**

I. der Spielmechanismus

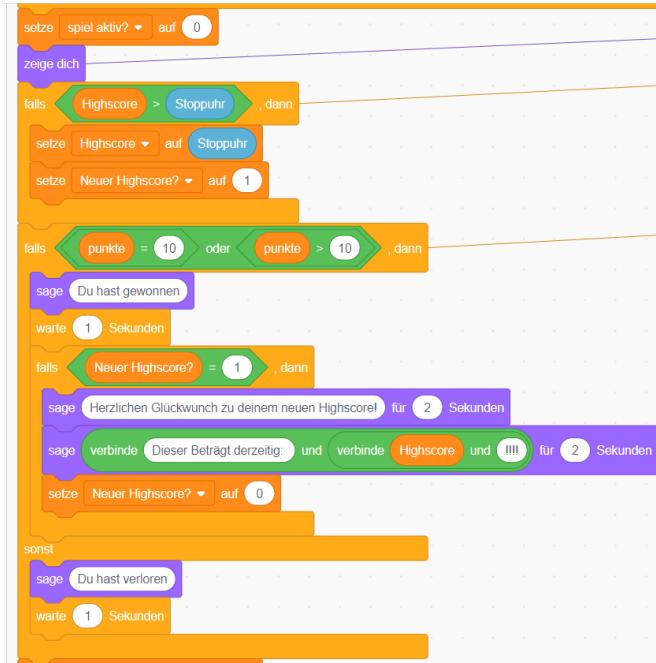


Sobald „spiel start“ empfangen wurde wird „Countdown“ gesendet und es wird gewartet bis der Script vom „Wenn Countdown empfangen wird“ beendet ist.

Wenn dies zutrifft, wird die Variable „spiel aktiv?“ auf 1 gesetzt, die Stoppuhr zurückgesetzt und der Definitionsblock „random xy“ mit den Argumenten für „speed“ = [0.5].

Der Definitionsblock wird so oft wiederholt, bis entweder die Stoppuhr über 15 [Sekunden] ist oder bis die Variable „punkte“ mindestens 10 beträgt. Sobald eine Bedingung erfüllt wurde geht es zu II.

II. die Highscore-Berechnung und die Bestimmung von Sieger/Verlierer



Die Variable „spiel aktiv?“ wird auf 0 gesetzt und die Figur zeigt sich.

Dann checkt sie ob die Zeit schneller war als der lokale Highscore, wenn ja dann wird der Highscore aufs höhere geändert und die Variable „Neuer Highscore?“ auf 1 gesetzt

Danach wird geguckt ob der Spieler gewonnen hat, indem gecheckt wird ob Punkte \geq [größer | | gleich] 10.

Falls dies zutrifft, dann sagt die Figur „Du hast gewonnen“ und wartet eine Sekunde.

Dann checkt es ob ein neues Highscore vorhanden ist, wenn ja dann gratuliert die Figur den Spieler, nennt den neuen Highscore und setzt die Variable „Neuer Highscore?“ auf 0.

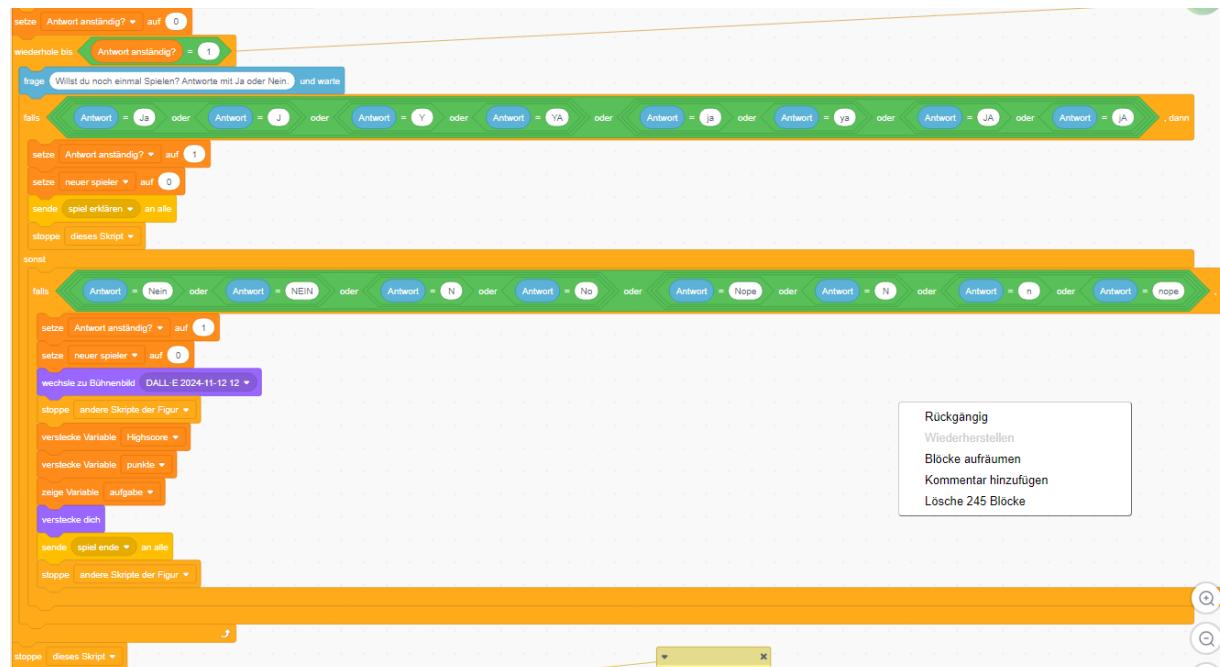
Highscore?“ auf 0.

Wenn die Punkte aber nicht größer| |gleich 10 ist, dann sagt die Figur „Du hast verloren“.

Der Highscore wurde zuerst berechnet, und nicht wer gewonnen hat, damit man eine akkurate Zeit bekommt.

Dann geht es rüber zu III.

III. Die Frage um nochmal zu spielen



Rückgängig
Wiederherstellen
Blöcke aufräumen
Kommentar hinzufügen
Lösche 245 Blöcke

In diesen Blockteil wird erst die Variable „Antwort anständig“ auf 0 gesetzt.

Dann wird gefragt ob der Spieler noch mal spielen will, falls die Antwort Ja [oder ähnliches] ist wird Antwort anständig auf 1 und neuer Spieler auf 0 gesetzt.

Zusätzlich wird „Spiel erklären“ an alle gesendet und dieser Script wird gestoppt

Falls die Antwort Nein [oder ähnliches] ist, dann werden die Variable „Antwort anständig?“ auf 1 und „neuer Spieler“ auf 0 gesetzt.

Dann wird zum Bühnenbild „DALL-E...“ gewechselt, die anderen Scripte der Figur gestoppt, die Variablen „Highscore“ und „punkte“ versteckt, „Aufgabe“ wird wieder gezeigt, die Figur versteckt sich, es wird „spiel ende“ gesendet und dieser Script wird dann gestoppt.

Figur TXT

Funktion

Diese Figur dient und den Spieler den Start des Spieles anzugeben.

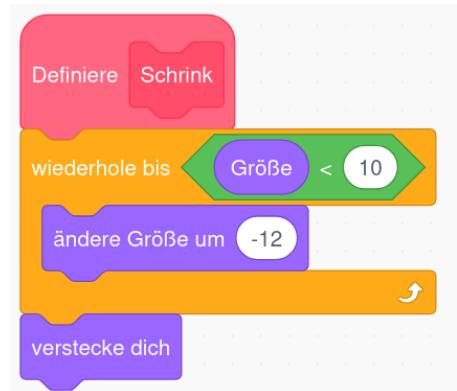
Startblock



Wenn das Programm startet dann versteckt sich die Figur und es geht zu x:0 und y:0.

Definition

„Schrink“



Dieser Definitionsblock ist da um die Figur magisch verschwinden zu lassen, indem die Figur kleiner und kleiner wird.

Wenn es die Größe der Figur unter 10 ist dann kommt es aus dem Wiederholen raus und versteckt sich dann.

Empfangsblöcke

Countdown



Wenn das Befehl „Countdown“ empfangen wird, geht diese Figur zur vordersten Ebene, setzt die Größe auf 100, wechselt zum Kostüm „3“ und zeigt sich.

Es funktioniert wie eine Art Countdown, weil in den folgenden Sekunden die Figur zum Kostüm „2“, dann „1“ und danach „Los!“.

Dann führt es den Definitionsblock „Schrink“ aus und verschwindet, nach den Countdown.

Kostüme

3

Kostüm „3“

2

Kostüm „2“

1

Kostüm „1“

LOS!

Kostüm „Los!“

Figur Wizard

Funktion

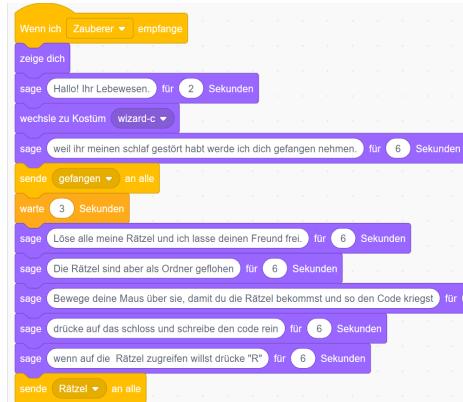
Diese Figur ist ein Teil von der Lösung von Aufgabe 7. Die Figur beinhaltet die Erklärung zum Lösen des Rätsels.

Wenn das Projekt startet – Block



Wenn das Projekt startet dann wird „rätsel angezeigt“ auf 0 gesetzt, die Variablen „rätsel angezeigt“ und „richtig?“ werden versteckt, die Figur versteckt sich und wechselt zum Kostüm B. Dann geht die Figur zu x:-187 und y: -56, die Richtung wird auf 90 Grad gesetzt und die Größe wird auf 50 [Einheiten] gesetzt.

Wenn „Zauberer“ empfangen wird – Block



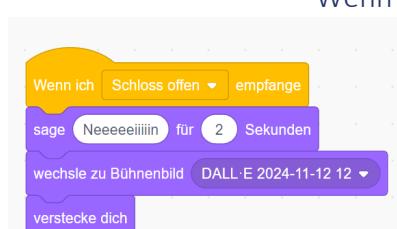
Sobald Zauberer empfangen wird, zeigt sich die Figur und sagt: „Hallo! Ihr Lebewesen.“

Dann sagt die Figur: „weil ihr meinen schlaf gestört habt werde ich dich gefangen nehmen.“ und sendet an alle gefangen. 3 Sekunden wird gewartet und die Figur sagt dann:

„Löse alle meine Rätsel und ich lasse deinen Freund frei.
Die Rätsel sind aber als Ordner geflohen. Bewege deine Maus über sie, damit du die Rätsel bekommst und so den Code kriegst. Drücke anschließend auf das Schloss und schreibe den Code rein. Wenn du auf die Rätsel zugreifen willst, drücke R“

Anschließend wird „Rätsel“ an alle gesendet.

Wenn „Schloss offen“ empfangen wird – Block



Sobald das Schloss offen ist, sagt die Figur „Neeeeeeeiiiiin“, das Bühnenbild wird zu „DALL-E...“ geändert und die Figur versteckt sich.



Wenn „r“ gedrückt wird – Block

Wenn „r“ gedrückt wird, checkt das Script ob die Variable „rätsel angezeigt“ gleich 1 ist.

Trifft dies zu, dann wird die Liste „Rätsel“ versteckt und „rätsel angezeigt“ wird auf 0 gesetzt.

Trifft dies nicht zu, dann wird die Liste „Rätsel“ gezeigt und „rätsel angezeigt“ wird auf 1 gesetzt.

Am Ende wird dieser Script gestoppt.

Dieser Script dient als Schalter [toggle] um den Rätsel anzuzeigen und zu verstecken.

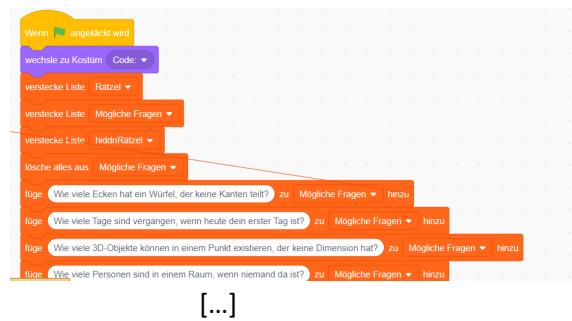
Figur Schloss

Funktion

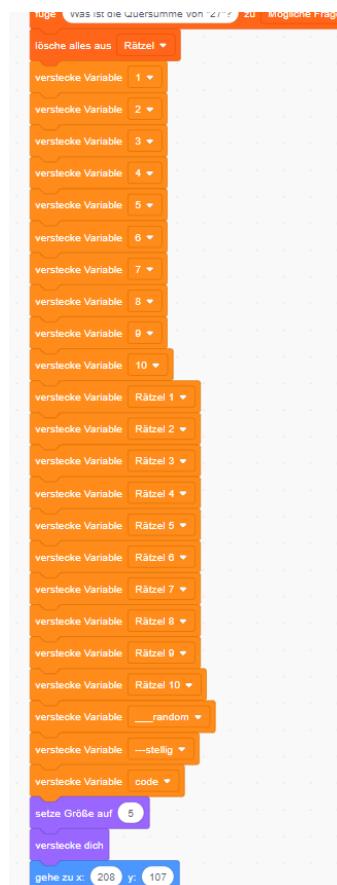
Diese Figur ist teil der Lösung von Aufgabe 7. Sie generiert die Fragen und den Code und fragt nach den Code.

Wenn man die Rätsel richtig gelöst hat, dann wird diese Figur den Schloss öffnen.

Startblock



Sobald das Projekt startet, wechselt die Figur zum Kostüm „Code:“, die Listen „Rätsel“, „hiddenRätsel“ und „Mögliche Fragen“ zu verstecken. Die Liste „Mögliche Fragen“ wird geleert und dann werden alle 50 Fragen zu „Mögliche Fragen“ hinzugefügt.



Dann wird alles aus „Rätsel“ gelöscht und die Variablen 1 bis 10, Rätsel 1 bis 10, __random, --stellig sowie code werden versteckt.

Dann wird die Größe der Figur auf 5 [Einheiten] gestellt, die Figur versteckt sich und die Figur geht zu x:208 und y:107.

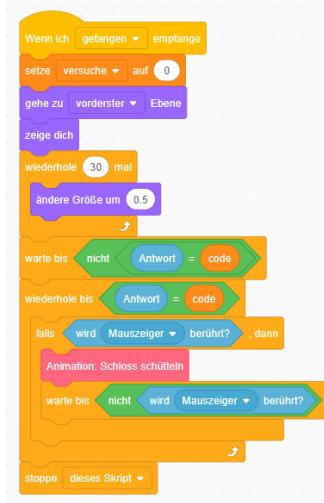
Wenn „Rätsel“ empfangen wird – Block



Sobald „Rätsel“ empfangen wird, wird n-mal ein Klon von der Figur Rätsel gemacht. Die Zahl n steht für die Stellenanzahl des Codes (z.B. n= 5 wenn die Zahl

[abcde] ist. Die Buchstaben a,b,c,d und e sind eine Zahl von 1-9). Dann wartet das Script bis die Länge von Rätsel doppelt so groß ist wie vom Wert der Variable ---stellig.

Wenn „gefangen“ empfangen wird (I) – Block



Sobald „gefangen“ empfangen wird, wird „versuche“ auf 0 gesetzt, die Figur geht zur obersten Ebene, zeigt sich und ändert seine Größe mit 0.5er [Einheiten] um 15 [Einheiten].

Dann wird gewartet bis die Antwort nicht gleich code ist.

Danach wird wiederholt bis die Antwort gleich der Code ist. In der Wiederholung ist falls-Block.

Dieser sagt, wenn die Figur vom Mauszeiger **berührt** wurde, dann führt es „Animation: Schloss schütteln“ aus und wartet bis es nicht mehr berührt wird.

Sobald die Bedingung [Antwort = Code] wahr ist wird dieser Script gestoppt.

Wenn die Figur angeklickt wird – Block



Wenn diese Figur angeklickt wird, dann wird gecheckt ob man die drei Versuche schon benutzt hat oder nicht.

Wenn man sie benutzt hat, dann werden alle andere Scripte dieser Figur gestoppt und es die Figur sagt, dass der Code 3 Mal falsch angegeben ist. Und dann wird alles gestoppt.

Wenn die Versuche noch nicht benutzt sind, dann wird gefragt was der Code ist. Wenn die Antwort richtig ist, dann wird „plopp“ gespielt, die Figur sagt richtig, sie verkleinert mit Verzögerung die Größe um -25 [Einheiten], die Figur versteckt sich dann und sendet an alle Schloss offen und stoppt alle andere Scripte dieser Figur.

Wenn der Code falsch angegeben ist, dann sagt die Figur Falsch und ändert die Variable „versuche“ um 1.

Am Ende wird dieser Script gestoppt.

Wenn „Animation: Schloss schütteln“ empfangen wird – Block



Wenn das empfangen wird, dann wird „Animation: Schloss schütteln“ ausgeführt.

Dieses Script wird von der Figur Winki gebraucht.

Wenn „gefangen“ empfangen wird (II) – Block



Wenn „gefangen“ empfangen wird, dann wird erst „Stellen berechner“ ausgeführt und dann „Code rechner && Rätsel ersteller“.

Definitionsblöcke

„stellen berechner“



Um den Code zu sehen → es ist in der Figur Schloss unter „Definiere Stellenberechner“

Sobald dieser Definitionsblock ausgeführt wird, wird die Variable „---stellig“ auf 0 gesetzt und „__random“ auf eine Zufallszahl von 1-100.

Der größere Block bestimmt die Stellenanzahl des Codes [---stellig=___] und ändert den Kostüm zu den zugeordneten Stellenwerten. Zum Beispiel 7 dann wird das Kostüm auf „Code:7“ ändert.

Die Wahrscheinlichkeiten werden im Folgenden Format aufgelistet:

{Stellenanzahl}: {Wahrscheinlichkeit}

1: 0%

2: 0%

3: 0%

4: 2%

5: 5%

6: 20%

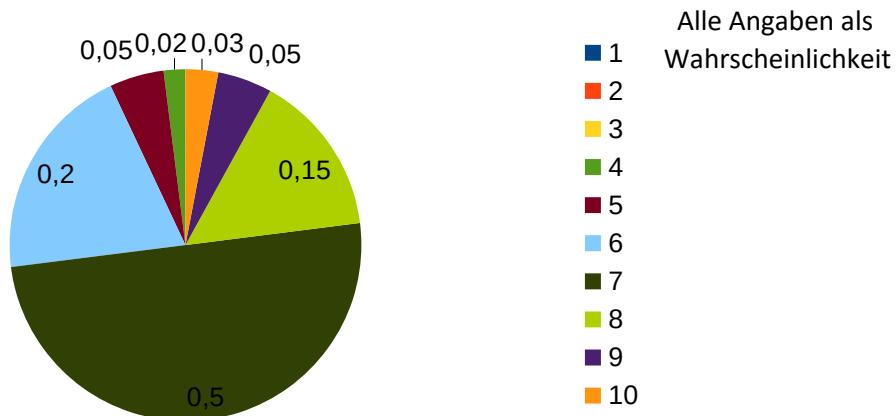
7: 50%

8: 15%

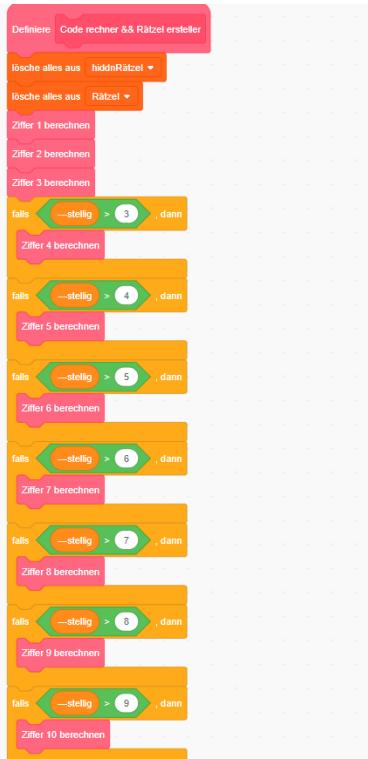
9: 5%

10: 3%

Oder auch als Diagramm



„Code rechner && Rätsel ersteller“



Dieser Definitionsblock löscht alles aus den Listen „hiddenRätsel“ und „Rätsel“.

Dann führt es die Definitionsblöcke Ziffer 1-3 berechnen.

Danach werden die Definitionsblöcke Ziffer 4-10 berechnen nur ausgeführt, wenn die Variable „---stellig“ = 10 ist.

Eigentlich werden die Definitionsblöcke Ziffer 1 bis n berechnen, wobei n die Variable „---stellig“ darstellt.

„Ziffer 1 berechnen“

„Die erste Ziffer:“ wird zu „hiddenRätsel“ hinzugefügt.

Dann wird die Variable 1 auf eine Zufallszahl von 0-9 gesetzt

Danach wird die Variable Rätsel 1 auf eine Zufallszahl von x^5+1 bis x^5+5 . X ist die Variable 1.

X wird mit 5 multipliziert, weil es 5 Fragen gibt und dann wird 1 oder 5 dazu addiert.

Die 1 und die 5 dienen als Bereichsbestimmung. Hier ist auch eine Tabelle dazu.

Zuteilung für die Ziffern im Code:

Zifferl: Rätsel (Anfang bis Ende

0: 1-5,
1: 6-10,
2: 11-15,
3: 16-20,
4: 21-25,
5: 26-30,
6: 31-35,
7: 36-40,
8: 41-45,
9: 46-50

Nach dem die Zahl bestimmt wurde, wird das Element X von „Möglichen fragen“ zu „hiddnRätsel“ kopiert. X steht für die bestimmte Zahl

„Ziffer 2 berechnen“

„Die zweite Ziffer: “ wird zu „hiddnRätsel“ hinzugefügt.

Dann wird die Variable 2 auf eine Zufallszahl von 0-9 gesetzt

Danach wird die Variable Rätsel 2 auf eine Zufallszahl von $x*5+1$ bis $x*5+5$. X ist die Variable 2.

Ein „Wiederhole bis“ – Block verhindert, dass die vorherigen Rätsel gleich sind (also Rätsel 1 \neq Rätsel 2).

In den „Wiederhole bis“ – Block werden die Variablen 2 und Rätsel 2 genau so wie vorhin neu bestimmt.

Sobald die Bedingung vom „Wiederhole bis“ erfüllt wurde wird das Element X von „Möglichen fragen“ zu „hiddnRätsel“ kopiert. X steht für die Variable Rätsel 2.

Anschließend wird „Die dritte Ziffer: “ zu „hiddnRätsel“ hinzugefügt.

„Ziffer 3 berechnen“

Die Variable 3 wird auf eine Zufallszahl von 0-9 gesetzt

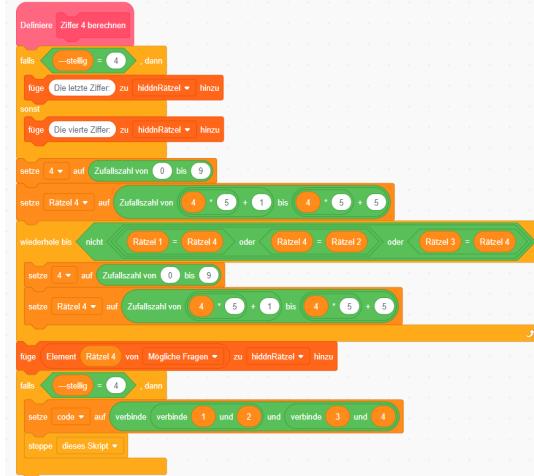
Dann wird die Variable Rätsel 3 auf eine Zufallszahl von $x*5+1$ bis $x*5+5$. X ist die Variable 3.

Danach verhindert „Wiederhole bis“ – Block, dass die vorherigen Rätsel gleich sind (also Rätsel 1 \neq Rätsel 2 \neq Rätsel 3).

In den „Wiederhole bis“ – Block werden die Variablen 3 und Rätsel 3 genau so wie vorhin neu bestimmt.

Sobald die Bedingung vom „Wiederhole bis“ erfüllt wurde wird das Element X von „Möglichen fragen“ zu „hiddnRätsel“ kopiert. X steht für die Variable Rätsel 3.

„Ziffer 4 berechnen“



Falls ---stellig gleich 4 ist, dann wird „Die letzte Ziffer:“ wird zu „hiddnRätsel“ hinzugefügt. Sonst wird „Die vierte Ziffer hinzugefügt.“

Dann wird die Variable 4 auf eine Zufallszahl von 0-9 gesetzt

Danach wird die Variable Rätsel 4 auf eine Zufallszahl von x^5+1 bis x^5+5 . X ist die Variable 4.

Ein „Wiederhole bis“ – Block verhindert, dass die vorherigen Rätsel gleich sind (also Rätsel 1 ≠ Rätsel 2 ≠ Rätsel 3 ≠ Rätsel 4).

In den „Wiederhole bis“ – Block werden die Variablen 4 und Rätsel 4 genau so wie vorhin neu bestimmt.

Sobald die Bedingung vom „Wiederhole bis“ erfüllt wurde wird das Element X von „Möglichen fragen“ zu „hiddnRätsel“ kopiert. X steht für die Variable Rätsel 4.

Anschließend wird nochmal gecheckt ob ---stellig gleich 4 ist.

Wenn dies zutrifft, dann wird die Variable code aus der Verbindung von den Variablen 1,2,3 und 4 erstellt.

Wenn dies nicht zutrifft, dann geht der Script weiter

„Ziffer 5-10 berechnen“

Die Blöcke werden zusammengefasst, weil sie identisch sind.



Falls ---stellig gleich n ist, dann wird „Die letzte Ziffer:“ wird zu „hiddnRätsel“ hinzugefügt. Sonst wird „Die n-te Ziffer“ hinzugefügt. Außer bei der Berechnung der zehnten Ziffer, denn da wird nur „Die letzte Ziffer“

Dann wird die Variable n auf eine Zufallszahl von 0-9 gesetzt

Danach wird die Variable Rätsel 4 auf eine Zufallszahl von x^5+1 bis x^5+5 . X ist die Variable n.

Ein „Wiederhole bis“ – Block verhindert, dass die vorherigen Rätsel gleich sind (also Rätsel 1 \neq Rätsel 2 \neq Rätsel 3 \neq Rätsel 4 \neq [...] \neq Rätsel (n-1) \neq Rätsel n).

In den „Wiederhole bis“ – Block werden die Variablen n und Rätsel n genau so wie vorhin neu bestimmt.

Sobald die Bedingung vom „Wiederhole bis“ erfüllt wurde wird das Element X von „Möglichen fragen“ zu „hiddnRätsel“ kopiert. X steht für die Variable Rätsel n.

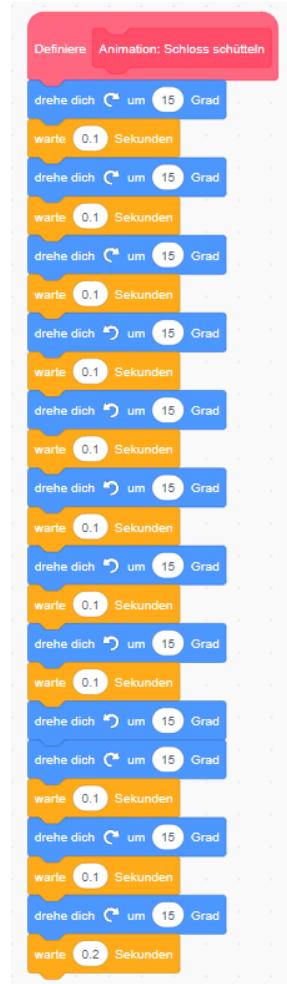
Anschließend wird nochmal gecheckt ob ---stellig gleich n ist.

Wenn dies zutrifft, dann wird die Variable code aus der Verbindung von den Variablen 1,2,3,4,[...],n-1 und n erstellt.

Wenn dies nicht zutrifft, dann geht der Script weiter.

n steht für die ...te Ziffer, die berechnet wird. Zum Beispiel bei der Berechnung von der 8ten Ziffer würde n = 8 gelten.

„Animation: Schloss schütteln“



Die Figur dreht sich nach rechts, dann mit Verzögerung nach links, dann wieder kurz nach rechts und dann links bis zur Ursprungsauslage.

Kostüme



Figur Rätsel

Funktion

Diese Figur ist teil der Lösung von Aufgabe 7. Diese Figur beinhaltet die Rätsel, die man braucht um den Code zu bekommen.

Definitionsblock

„random xy {speed}“

Der Definitionsblock random xy setzt die Größe der Figur aus einer Zufallszahl im Intervall [40;50], setzt die x-Koordinate aus einer Zufallszahl im Intervall [-219;207] und die y-Koordinate aus einer Zufallszahl im Intervall [-156;106]. Dann zeigt sich die Figur und es werde X Sekunden gewartet.

X ist das Argument welches beim Block eingegeben werden kann. Je kleiner X ist, desto schneller bewegt sich die Figur.



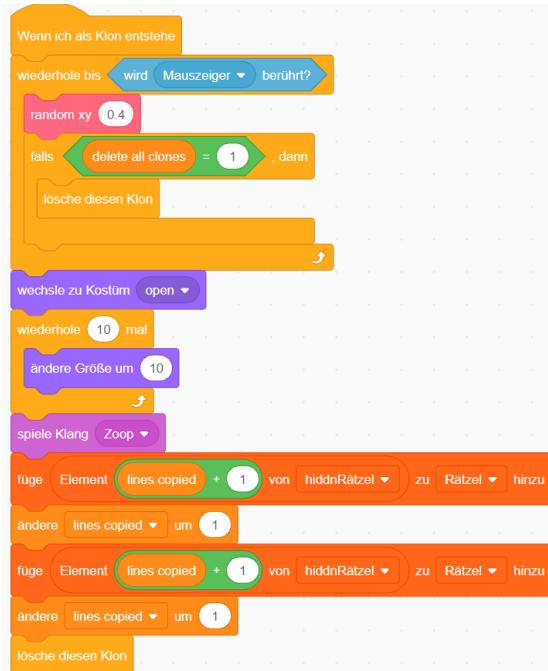
Startblock

Sobald das Projekt startet, wird die Größe auf 40 [Einheiten] gesetzt, das Kostüm auf "Folder" gewechselt und die Figur versteckt sich.

Dann werden die Variablen „lines copied“ und „delete all clones“ auf 0 gesetzt.



Wenn ich als Klon entstehe – Block



Sobald dieses Block ausgeführt wurde, wird random xy mit den Argument (0.4) ausgeführt.

Dann wird gecheckt ob „delete all clones“ gleich 1 ist. Wenn dies zutrifft, dann wird der Klon gelöscht.

Das Ganze wird wiederholt bis der Klon vom Mauszeiger berührt wird.

Nachdem die Bedingung erfüllt wurde, wechselt der Klon ins Kostüm „open“. Dann wird die Größe der Figur mit 10er [Einheiten] Verzögerung um 100 [Einheiten] geändert. Es wird dann „Zoop“ abgespielt.

Danach wird das Element (k+1) von „hiddnRätsel“ zu „Rätsel“ hinzugefügt.

Dann wird k um 1 erhöht und das Element (k+1) von „hiddnRätsel“ zu „Rätsel“ hinzugefügt.

Anschließend wird k um 1 erhöht und der Klon wird gelöscht.

K ist die Variable „lines copied“.

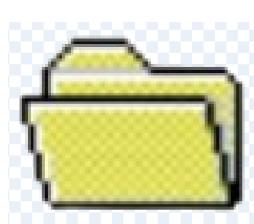
Es wird zwei mal kopiert, weil wenn die Stellenanzahl 5 beträgt, die Anzahl von den Inhalt 2^* die Stellenanzahl ist.

Der Format von der Liste ist:

1. | Die erste Ziffer |
2. | [...] |
3. | Die zweite Ziffer |
4. | [...] |
5. [...]

usw.

Kostüme



Folder

open

Bühnenbild

Scripte

Startblock

Wenn das Projekt startet, dann wechselt das Bühnenbild zu „DALL-E...“



Bühnenbilder

DALL·E 2024-11-12 12



Moon



Woods



Variablen

(Es werden nur die Variablen hinzugefügt., die nicht schon in der Quelldatei waren!)

Variable (Figur): Funktion

code: ist der Code für den Rätsel

delete all clones: ist dafür da damit man alle klonen von Rätsel löschen kann

Highscore: (selbsterklärend)

lines copied: wie viele Zeilen vom versteckten Rätsel ins Rätsel kopiert werden

Name: Spielername, nach Eingabe des Spieles

neuer Spieler: Variable um zu gucken ob es ein neuer Spieler ist, oder ob nochmal gespielt werden will

spiel aktiv?: (selbsterklärend)

gefangen (Winki): Zeigt an ob Winki gefangen ist oder nicht

klick (Magic Wand): Zeigt wie oft der Zauberstab geklickt wurde

Antwort anständig? (Button1): Dient dazu um anzuzeigen ob mit Ja/Nein geantwortet wurde oder etwas anderes.

Neuer Highscore? (Button1): Dient als Indikator um zu gucken ob ein neuer Highscore da ist.

Punkte (Button1): Dient als Indikator um darzustellen wie oft die Figur Button 1 angeklickt wurde.

S gedrückt?: (selbsterklärend) ob s schon gedrückt wurde

rätsel angezeigt (Wizard): Dient als Indikator für das Script um zu checken ob die Liste Rätsel schon offen ist.

___random (Schloss): Zufallszahl

---stellig (Schloss): Stellenanzahl des Wertes von der Variable Code

1 (Schloss): Erste Ziffer vom Code

2 (Schloss): 2te Ziffer vom Code

3 (Schloss): 3te Ziffer vom Code

4 (Schloss): 4te Ziffer vom Code

5 (Schloss): 5te Ziffer vom Code

6 (Schloss): 6te Ziffer vom Code

7 (Schloss): 7te Ziffer vom Code

8 (Schloss): 8te Ziffer vom Code

9 (Schloss): 9te Ziffer vom Code

10 (Schloss): 10te Ziffer vom Code

temp timer (Schloss): Dient als Debug

Rätsel 1 (Schloss): Erste Zahl vom Element aus mögliche Fragen

Rätsel 2 (Schloss): 2te Zahl vom Element aus mögliche Fragen

Rätsel 3 (Schloss): 3te Zahl vom Element aus mögliche Fragen

Rätsel 4 (Schloss): 4te Zahl vom Element aus mögliche Fragen

Rätsel 5 (Schloss): 5te Zahl vom Element aus mögliche Fragen

Rätsel 6 (Schloss): 6te Zahl vom Element aus mögliche Fragen

Rätsel 7 (Schloss): 7te Zahl vom Element aus mögliche Fragen

Rätsel 8 (Schloss): 8te Zahl vom Element aus mögliche Fragen

Rätsel 9 (Schloss): 9te Zahl vom Element aus mögliche Fragen

Rätsel 10 (Schloss): 10te Zahl vom Element aus mögliche Fragen

versuche (Schloss): Die Anzahl der Versuche beim Code eintippen

Listen

Die Liste Mögliche Fragen ist eine Liste mit 50 Fragen, deren Antwort von 0-9 sind.

Sie sind so angeordnet:

Zuteilung für die Ziffern im Code:

Zifferl: Rätsel (Anfang bis Ende

0: 1-5,

1: 6-10,

2: 11-15,

3: 16-20,

4: 21-25,

5: 26-30,

6: 31-35,

7: 36-40,

8: 41-45,

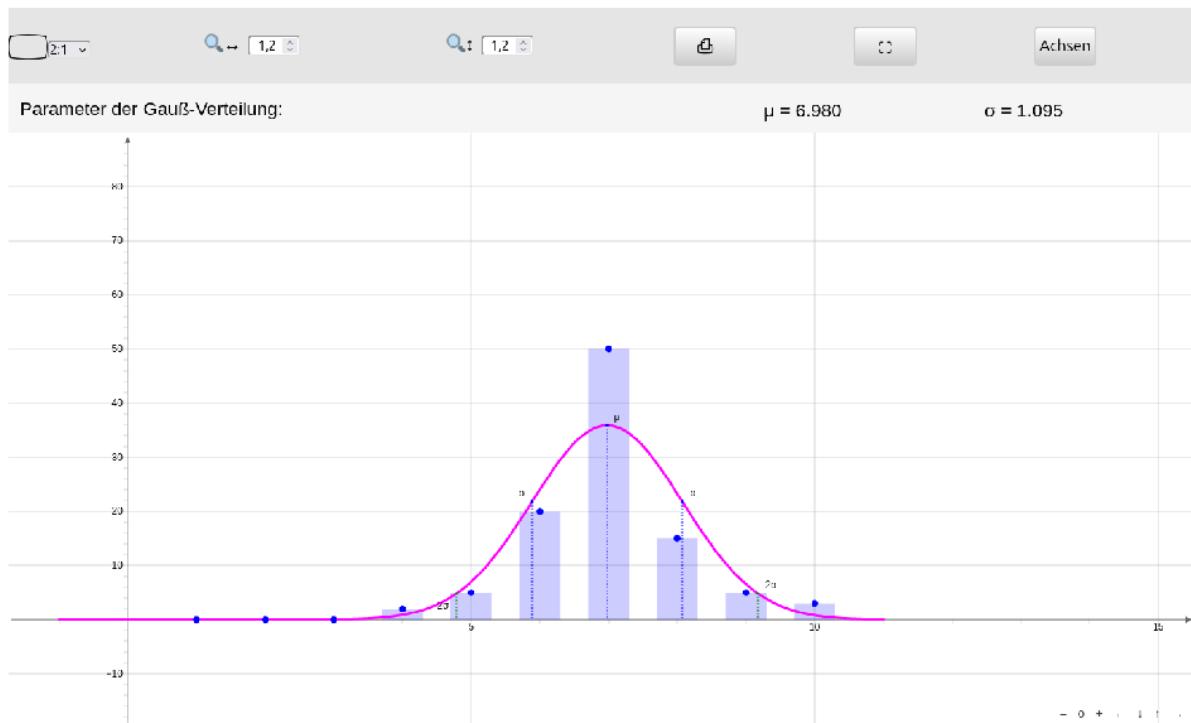
9: 46-50

Die hiddnRätsel sind die Rätsel die zum Code korrespondieren aber versteckt sind.

Die Rätsel ist nur eine Kopie von hiddnRätsel, aber man muss die Ordner fangen, damit sie kopiert werden. Diese werden angezeigt.

Zufallsverteilung

Approximation (fit) der Gauß Verteilung an Messwerte



Hier sind die Wahrscheinlichkeiten in der Normalverteilung.

Das Wert für den Erwartungswert μ ist 6.980 und das Wert für die Standardabweichung σ = 1.095.

Alle Werte beim Code sind Zufällig generiert, auch die Anordnung sowie die Auswahl der Rätsel sind Zufällig.