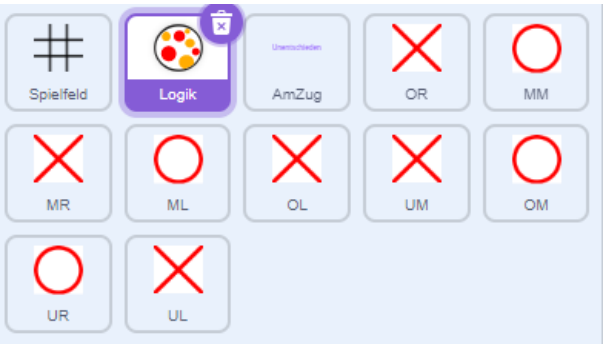
Rückblick: Als aller erstes haben wir festgestellt, dass wir zwei Situationen betrachten müssen: Entweder beginne ich oder der Gegner. Wir haben den Fall betrachtet, dass der Gegner beginnt und wir so spielen, dass wir nicht verlieren können. In unserer folgenden Implementierung verfolgen wir das Ziel, einen Computer dermaßen gut zu programmieren, so dass es ihm nicht mehr möglich ist, zu verlieren. Als Gegner wird der Spieler betrachtet. Auch in dieser Implementierung beginnt der Gegner – also der Spieler.

Es ist wichtig, dass verstanden wird, dass „wir“ der Computer sind. Der Gegner ist der Benutzer, der mit dem Computer spielt.

Bevor den Schülern und Schülerinnen das vorgefertigte Gerüst und die Schritt für Schritt Anweisung ausgehändigt wird, ist es wichtig, das Gerüst bzw. die einzelnen Figuren kurz zu erklären.



Es müssen nicht alle Blöcke der Figuren verstanden werden. Wichtige bzw. zu bearbeitende Blöcke sind mit Kommentaren versehrt und links oben in den Skripten der Figuren platziert.

Die Figur „Spielfeld“ ist für die Optik zuständig. Entweder wird ein Spielfeld mit oder ohne Gewinnsymbol angezeigt.

Die Figur „AmZug“ gibt Auskunft, wer im Moment am Zug ist und ist ebenfalls nur für die Optik zuständig.

Die Figur „Logik“ ist das Gehirn des Computers. Sie setzt die Kreuze für ihn. Wo die Kreuze gesetzt werden müssen, damit der Computer nicht verlieren kann, wurde bereits im analogen Einstieg erarbeitet.

Zu guter Letzt gibt es neun Felder. Für jedes Feld wurde eine eigene Figur angelegt. Klickt der Benutzer auf ein Feld oder beansprucht der Computer ein Feld für sich, ändert sich das Kostüm des Feldes dementsprechend. Die Figuren der Felder wurden nach folgendem Muster bezeichnet:

OL	OM	OR
ML	MM	MR
UL	UM	UR

Bevor nun die Arbeitsanleitung und das Gerüst ausgehändigt wird, ist wichtig zu betonen, dass im Gerüst nur Veränderungen vorgenommen werden dürfen, die in der Anleitung beschrieben werden.

Die Aufgabenstellung erfolgt in der Form von Kärtchen. Jede Aufgabe befindet sich auf einem eigenen Kärtchen und wird in vier Teile gegliedert. Links oben ist die zu bearbeitende Figur abgebildet. Rechts oben die Arbeitsanweisung, links unten Tipps und rechts die Musterlösung bzw. ein Teil der Musterlösung oder eine Zusatzaufgabenstellung.

Bevor das Gerüst und die Arbeitsanweisung ausgehändigt werden, muss sich die Lehrperson mit den Blöcken, die Kommentare besitzen, und der Musterlösung vertraut machen. (Beschreibung auf den nächsten Seiten).

Da den Schülern und Schülerinnen nur ein Gerüst ausgehändigt wird, funktioniert das Programm natürlich so schnell noch nicht. Nach jeder Aufgabe kann der Schüler oder die Schülerin allerdings einen Fortschritt erkennen.

Des Weiteren ist empfehlenswert, dass die bereits angelegten Variablen im Vorfeld mit den Schülern und Schülerinnen besprochen werden.

- amZug: 0 wenn SpielerIn an der Reihe, 1 wenn Computer an der Reihe
- name: eingegebener Name
- fall: leer zu Beginn. Nach erstem Kreuz „Rand“, „Mitte“ oder „Ecke“
- gesetzteKreuze: Anzahl der belegten Felder, zu Beginn 0, bei Unentschieden 9
- Weiters gibt es für jedes Feld eine Variable (siehe Abbildung). Diese Variablen können die Werte 0, 1, oder 2 annehmen. Zu Beginn 2, wenn der / die SpielerIn auf das Feld geklickt hat 0, wenn der Computer das Feld belegt hat 1

ol = 2	om = 2	or = 2
ml = 2	mm = 2	mr = 2
ul = 2	um = 2	ur = 2

Zuweisung zu Beginn

Aufgabe 1: Figur Logik (Namensgebung)

Zu Beginn muss der / die BenutzerIn den Namen eingeben und die Antwort der Variable „name“ zugewiesen werden. Es wurde bereits die Variablen amZug erstellt. Diese nimmt den Wert 0 an, wenn der /die BenutzerIn am Zug ist und den Wert 1 an, wenn der Computer am Zug ist. Da der /die BenutzerIn beginnt, wird der Wert 0 zugewiesen. Ändert sich die Variable amZug, so muss dies dem restlichen Programm mitgeteilt werden. Es wird ZugWechsel an alle gesendet.

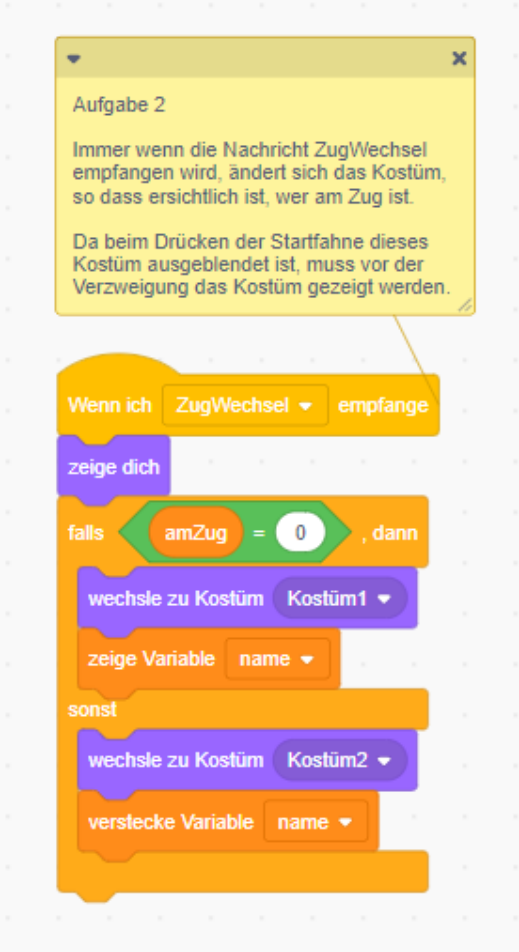
Wurde die Aufgabe erledigt, ist es möglich einen Namen einzugeben. Alles andere funktioniert noch nicht nach Wunsch.

Aufgabe 2: Figur AmZug (Zug-Logik)

Immer wenn die Nachricht ZugWechsel empfangen wird, ändert sich das Kostüm, so dass ersichtlich ist, wer am Zug ist. Da beim Drücken der Startfahne dieses Kostüm ausgeblendet ist, muss vor der Verzweigung das Kostüm gezeigt werden.

Falls amZug auf 0 gesetzt ist, ist der Spieler am Zug. Das Kostüm wechselt zu „Kostüm1“ und zeigt den Namen, den der / die Benutzerin bei Aufgabe 1 eingegeben hat.

Wird die Nachricht ZugWechsel gesendet und amZug ist gleich 1. Dann wird das Kostüm zu „Kostüm2“ gewechselt.



Wurde die Aufgabe erledigt, wird, nachdem der Name eingegeben wurde, zwar angezeigt, dass „name“ am Zug ist, allerdings kann der / die Benutzerin noch nicht überall ein Kreuz setzen.

Aufgabe 3 und 4: Figur MR und UR (SpielerInnen-Code)

Wir haben 9 Felder. Für diese wurden eigene Figuren und Variablen erstellt. Siehe Beschreibung oben. Klickt der / die SpielerIn auf ein Feld oder beansprucht der Computer eines für sich, ändern sich zum einen die Variablen und zum anderen die Skripte der Figuren. Dies funktioniert noch nicht bei den Figuren MR und UR. Diese sind von den Schülern und Schülerinnen zusammenzubauen.

Wenn der Spieler auf ein Feld ein Kreuz setzt, wird diese Variable auf 0 gesetzt. Belegt der Computer ein Feld, wird die Variable auf 1 gesetzt. Weiters sind für die Spielerlogik die Variablen fall und gesetzteKreuze wichtig. Wir haben bereits erarbeitet, dass es nur drei verschiedene Fälle gibt, mit denen der Spieler beginnen kann. Nachdem der Spieler sein erstes Kreuz gesetzt hat, wird die Variable fall auf Ecke, Rand oder Mitte gesetzt. Die Variable gesetzteKreuze ist zu Beginn des Spiels gleich null. Setzt der Spieler ein Kreuz oder der Computer ändert sich die Variable um 1. Für jedes Feld wurde auch eine Figur angelegt.

Bei Aufgabe 3 fehlen die Zuordnungen zu den Variablen. Bei Aufgabe 4 müssen die Schüler und Schülerinnen die Blöcke selbst bauen.

Wurden die Aufgaben erledigt, kann der / die SpielerIn überall Kreuze setzen. Der Computer antwortet allerdings noch nicht.

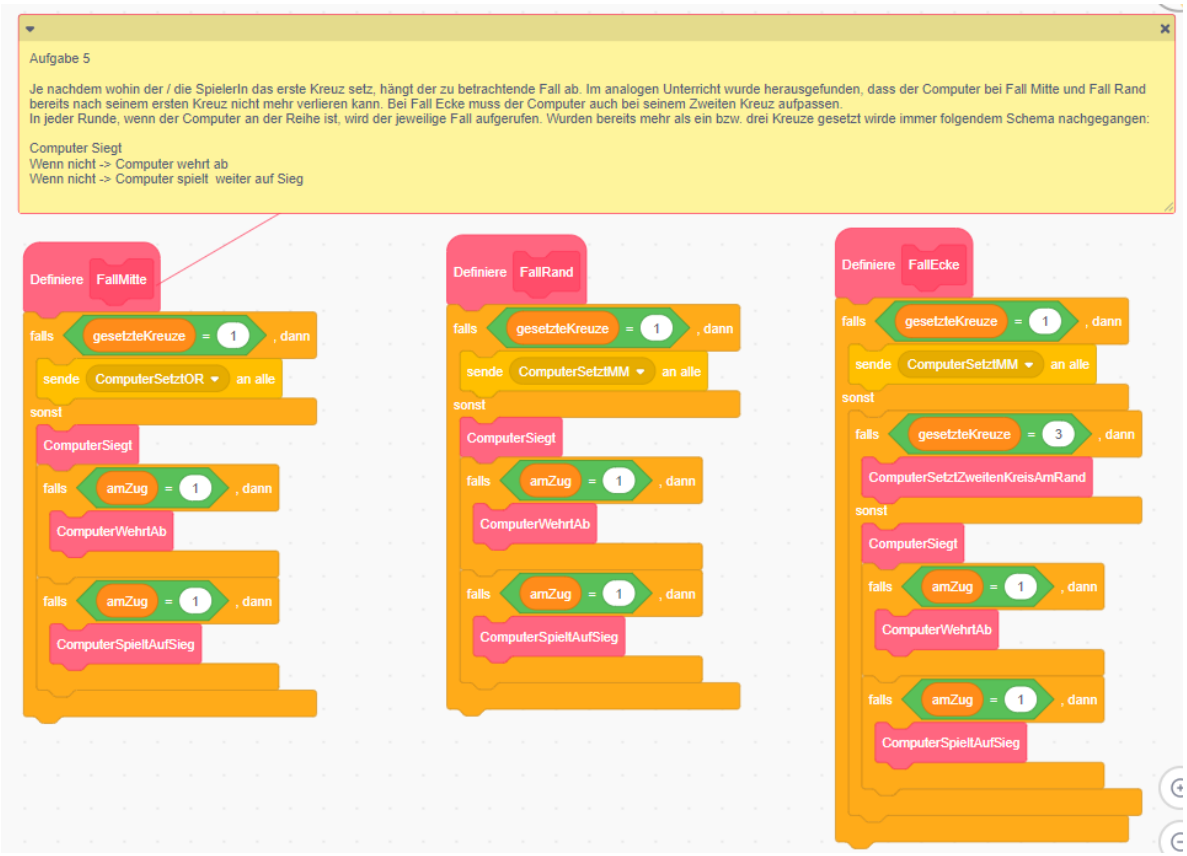
Aufgabe 5: Figur Logik (TicTacToeAlgorithmus)

Im analogen Einstieg wurde für die drei Fälle durch die Schüler und Schülerinnen bereits selbstständig herausgefunden, wie der Computer antworten muss, damit er nicht verlieren kann. Vergleicht man nun mit diesem erarbeiteten Pseudocode, kann die Computer-Logik schnell in Scratch implementiert werden.

Je nachdem wohin der / die SpielerIn das erste Kreuz setzt, hängt der zu betrachtende Fall ab. Im analogen Unterricht wurde herausgefunden, dass der Computer bei Fall Mitte und Fall Rand bereits nach seinem ersten Kreuz nicht mehr verlieren kann. Bei Fall Ecke muss der Computer auch bei seinem zweiten Kreuz aufpassen.

In jeder Runde, wenn der Computer an der Reihe ist, wird der jeweilige Fall aufgerufen. Wurden bereits mehr als ein bzw. drei Kreuze gesetzt wird immer folgendem Schema nachgegangen:

- Computer Siegt
- Wenn nicht -> Computer wehrt ab
- Wenn nicht -> Computer spielt weiter auf Sieg



Wurde die Aufgabe erledigt, antwortet der Computer auf das erste Kreuz. Der Computer kann jedoch noch nicht gewinnen und nicht drei Kreuze abwehren.

Aufgabe Z: Figur Logik (Computer-Logik) /Zusatz für leistungsstarke Schüler und Schülerinnen

Es wurden die Blöcke ComputerWehrtAb und ComputerSiegt definiert. Wenn der Spieler bereits zwei Kreuze nebeneinander hat, bleibt dem Computer nur an einer Stelle die Möglichkeit, den Sieg abzuwehren. Andererseits wenn der Computer bereits 2 Kreuze nebeneinander hat, muss er das nächste Kreuz an eine gewisse Stelle setzen. Der Computer setzt ein Kreuz, in dem die Figur Logik gewisse Nachrichten sendet. In den Blöcken ComputerWehrtAb und ComputerSiegt fehlen einige Nachrichten.

Allerdings muss der Computer in einer aufwendigen und mehrfachen Verzweigung alle Fälle kontrollieren. Bei komplizierten Programmstrukturen ist es oft hilfreich eine Skizze anzufertigen. In diesem Fall kann ein Spielfeld mit den Kreuzen und Kreisen erstellt werden. Dazu sollen die Variablen aus dem Gerüst verwendet werden.

Beispiel ComputerWehrtAb: In der ersten Abfrage wird der Fall betrachtet, ob (oben rechts und mitte rechts bereits ein Kreuz und unten rechts noch nichts ist) oder ob ein anderer Fall eintritt, so dass unten rechts abgewehrt werden muss

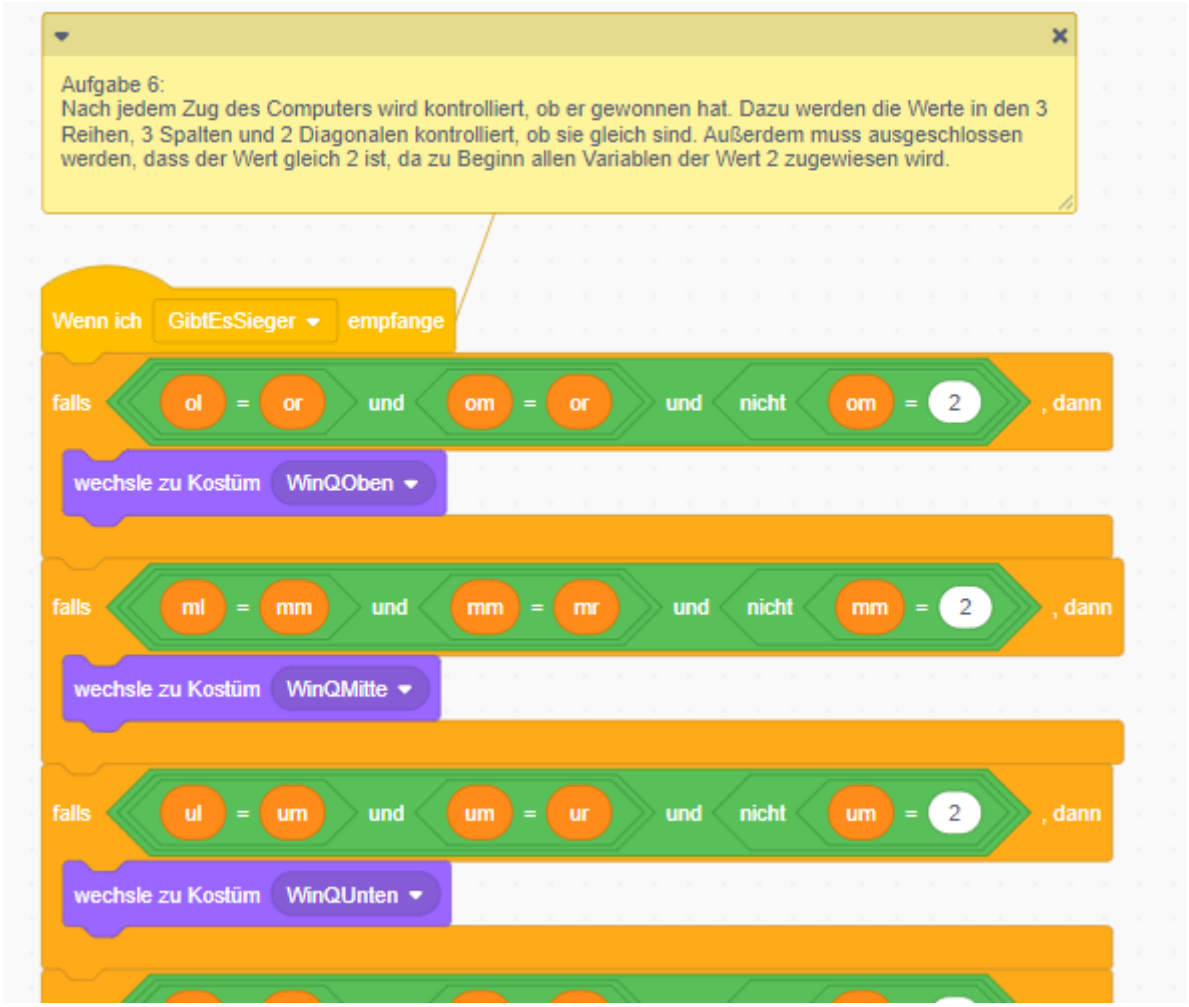
Beispiel ComputerSiegt: In der ersten Abfrage wird der Fall betrachtet, ob (oben rechts und oben mitte bereits ein Kreis und oben links noch nichts ist) oder ob ein anderer Fall eintritt, so dass mit oben links gewonnen werden kann.

Wurde diese Aufgabe erledigt, spielt der Computer bereits fehlerfrei. Es wird jedoch noch nicht ausgegeben, wenn jemand gewonnen hat.

Aufgabe 6: Figur Spielfeld (Winner Logik)

Wir haben bereits erarbeitet, dass es keine Gewinnmöglichkeit für den /die SpielerIn gibt, wenn der Computer fehlerfrei programmiert worden ist. Für den nächsten Auftrag sollen die Kostüme der Figur Spielfeld betrachtet werden. Sollte der Sieg in der oberen horizontalen Reihe gefeiert werden, wird zum Beispiel zu Kostüm WinQOben gewechselt.

Es wird verglichen, ob ol mit or und om mit or gleich sind. Somit wird garantiert, dass alle drei Felder dasselbe Kreuz besitzen. Allerdings garantiert uns das nicht, dass es einen Sieger gibt, da zu Beginn des Spiels alle Felder mit 2 definiert werden. Aus diesem Grund muss noch gecheckt werden, ob die die Werte nicht 2 sind.



Wurde die Aufgabe erledigt, funktioniert das Programm.