

Weg mit dem Weltraumschrott!

Nachdem alle Bilder heruntergeladen wurden, kannst du mit der Programmierung des Spiels anfangen.

Schritt 1:

- Öffne eine neue Scratch-Datei und speichere sie auf deiner Festplatte unter dem Dateinamen „weltraum-1“ ab
- Lösche die Scratchkatze aus den Figuren
- Jetzt erstellst du ein Geschoss, das im Weltraum den Müll abschießen soll. Da das Weltall dunkelblau ist, sollte das Geschoss sich von der Farbe absetzen.
- Wähle den Pinsel im Figurenmenü – du landest bei der Kostümdesignerfläche
 - Wähle die Kreisform
 - Die Füllfarbe soll gelb (Farbe 15, Sättigung 100, Helligkeit 100) sein.
 - Die Randfarbe ein dunkles Orange. (Farbe 6, Sättigung 100, Helligkeit 100, die Randdicke 8)
 - Ziehe einen Kreis auf
 - Nenne deine Figur Kugel
 - Wechsle zurück zum Skript deiner neuen Kugel



Schritt 2:

- Deine Kugel soll später im Weltraum hin- und herfliegen und den Schrott (olle Planeten, alte Raketen, Raumschiffe und alte Satelliten) abschießen.
- Deshalb musst du die Kugel entsprechend programmieren.

- Wenn die Kugel am Rand der Bühne ankommt, soll sie abprallen.
- Die Kugel muss die Richtung ändern
- Sie soll an einer bestimmten Position starten
- Die Bewegungen werden nicht gestoppt, es sei denn ...



Schritt 3

Ein Breakoutspiel funktioniert so, dass ein Gegenstand eine Kugel zurückwirft, um damit Gegenstände zu treffen. In unserem Spiel sollen die Gegenstände verschwinden, wenn die Kugel sie trifft. Der Gegenstand, der die Kugel zurückwirft, ist bei uns ein Raumschiff, das sich nur nach links oder rechts bewegen kann.

Jetzt ist es wichtig zu überlegen, ob das Spiel mit den Pfeiltasten oder mit der Maus und den Fingern (beim Touchscreen) gesteuert werden soll. Je nachdem, was du wählst, muss die Steuerung programmiert werden.

Das Raumschiff mit den Pfeiltasten steuern

- Lade das Raumschiff-4 in die Figurensammlung hoch
 - Ändere die Größe auf 40

- Ziehe den Ereignisblock mit der grünen Flagge in den Codebereich
- Wähle einen Startpunkt für das Raumschiff in der Mitte am unteren Bildschirmrand z. B. /-158. Das bewirkt, dass sich das Raumschiff bei jedem Start an der gleichen Stelle befindet
- Damit die Pfeiltastensteuerung flüssig läuft und nicht hakelt, nehmen wir folgenden Code.
- Da das Raumschiff sich nur nach links und rechts bewegen soll, aber nicht nach oben unten, müssen nur die linke und rechte Pfeiltaste programmiert werden.
- Um eine ununterbrochene Steuerung zu garantieren, muss ständig abgefragt werden, ob die Tasten gedrückt werden. (wiederhole fortlaufend)



Das Raumschiff mit der Maus oder den Fingern (am Touchscreen) steuern

Wenn du das Raumschiff lieber mit der Maus steuern willst oder auf dem Tablet mit den Fingern, musst du eine andere Steuerung einbauen.

- Ziehe den Block „Wenn Fahne geklickt“ in den Codebereich
- Dann brauchst du wieder den Block „wiederhole fortlaufend“.
- Jetzt brauchst du einen Block, der die Bewegung auf der X-Achse kontrolliert. Nimm „setze X auf“.
- Nun wähle aus der „Fühlen-Palette“ den Block „Maus-X-Position“ und platziere ihn in dem Zahlenbereich. Statt einer festen Zahl wird so jetzt immer die aktuelle Maus-X-Position verwendet.



Eventuell hast du jetzt beide Steuerprogramme in einem Codebereich und möchtest deinen Programmcode flexibel halten. Leider geht das nicht. Man kann keine Tastatursteuerung und Maussteuerung in einem einzigen Programmcode anbieten. Das ist wirklich schade.

Was kannst du jetzt machen, wenn du beide Programme behalten willst?

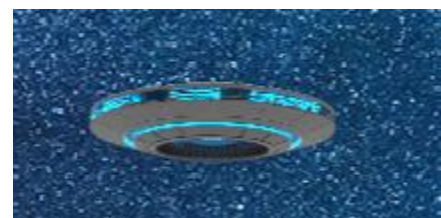
Du kannst von dem Steuerprogramm, das du zwar behalten willst, aber nicht zum Spielen brauchst, den Startblock „Wenn Flagge angeklickt“ löschen. Dann wird dieser Block nicht ausgeführt und nur das andere Steuerprogramm wird vom Spiel genutzt.

Wenn du nicht beide Steuerprogramme behalten willst, lösche das, was du nicht brauchst aus dem Codebereich.

Schritt 4:

Es wird Zeit für ein Bühnenbild

- Wähle das Bild „sternenhintergrund-480-x-x360px.jpg“ aus und lade es in die Bühnenbilder.
- Schon steht das Raumschiff im Weltall



Es wird Zeit dein Projekt zu speichern!

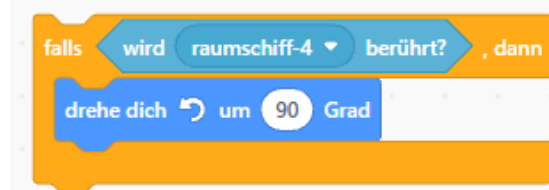
Schritt 5:

Wenn du nun auf die grüne Flagge klickst, kannst du dein Raumschiff bewegen und die Kugel fliegt durch die Gegend und prallt immer vom Rand ab. Allerdings fliegt sie durch das Raumschiff hindurch als wäre es gar nicht vorhanden.

Das muss jetzt geändert werden. Jedes Mal, wenn die Kugel das Raumschiff trifft, soll sie auch vom Raumschiff abprallen und wieder in die Weite des Weltalls zurückgeschleudert werden.

Aber wie funktioniert das? Mit dem Block „Pralle vom Rand ab“ wird es nicht klappen. Das Raumschiff ist schließlich kein Rand. Aber es gibt natürlich eine Lösung – wir müssen sie nur selbst programmieren.

- klicke auf die Kugel im Figurenbereich. Jetzt siehst du das bisherige Programm der Kugel. 
- Füge eine Abfrage hinzu, in der festgestellt werden soll, ob die Kugel das Raumschiff berührt.
- Wenn ja, soll sich die Kugel um 90 Grad nach links drehen und zurück in den Weltraum fliegen.
- Jetzt müssen wir das Codefragment in den Code der Kugel einfügen. Das Ganze sieht dann so aus:
→



Jetzt wird es Zeit, das Ganze mal auszuprobieren. Speichere dein Programm und klicke auf die grüne Fahne.

Was fällt dir auf? Ist das Raumschiff zu klein oder die Kugel zu schnell?

Dann ändere die entsprechenden Werte, so wie du sie gut findest.

Wie findest du dein Spiel bis jetzt?

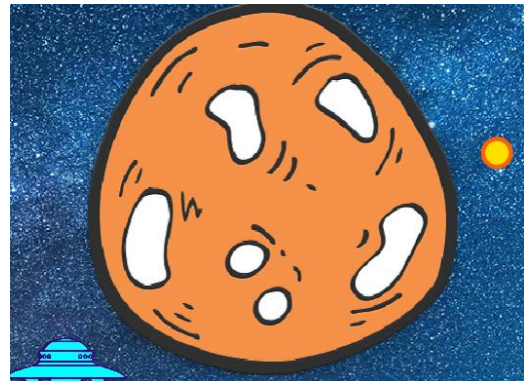
Langweilig?

Klar, jetzt muss noch der Weltraumschrott platziert werden, der dann von der Kugel des Raumschiffs abgeschossen werden kann. Das macht das Spiel schon spannender.

Schritt 6:

Der Weltraumschrott (alle Planeten, alte Raketen, Raumschiffe und alte Satelliten) müssen noch im Weltall verteilt werden.

- Lade zuerst die ollen Planeten in die Figuren hinein.
- Wie du sehen kannst, sind sie viel zu groß. Deshalb musst du im nächsten Schritt die Größe der Figuren ändern. Damit du nicht den Überblick verlierst, kannst du die Planeten, die du gerade nicht bearbeitest, für den Moment unsichtbar machen.
- So sieht die Bühne dann aus. Jetzt muss der Planet verkleinert werden.
- Damit alle Planeten am Ende in etwa gleich groß sind, wäre 10 eine gute Größe.
- Jetzt haben die Kugel und der Planet-1 in etwa die gleiche Größe.



- Verkleinere die anderen Planeten auch so weit, dass sie ungefähr die gleiche Größe wie die Kugel haben. Dazu musst du die Planeten auch wieder sichtbar machen 😊
- Jetzt sind die Satelliten dran und zum Schluss die alten Raketen und das Raumschiff
- Verteile alle Schrottteile und lass etwas Platz zwischen den ihnen, damit Scratch nachher alle Treffer sauber zählen kann. Liegen die Schrottteile zu dicht beieinander, kann die Kugel mehrere fast gleichzeitig treffen und es wird eventuell nicht jeder Punkt gezählt. Platziere die Schrottteile aber auch nicht ganz in den Ecken, sonst wird das Spiel unnötig schwierig.
- So sieht der Schrottplatz im Weltraum aus. Gut, oder?



Schritt 7

Jetzt braucht der Schrott aber auch noch ein Programm, das ihm sagt, was er machen soll, wenn er von der Kugel berührt wird.



- Klicke auf Planet-1

- Was soll passieren, wenn die Kugel den Planeten berührt? Richtig, er soll verschwinden (und später gibt es dann auch einen Punkt dafür – aber das kommt noch 😊)
- Also brauchen wir eine fortlaufende Abfrage (Schleife), die fragt, ob die Kugel vom Planeten berührt wurde.
- Wenn das der Fall ist, soll sich der Planet verstecken. Das bedeutet, dass er am Anfang des Spieles sich auch zeigen soll.

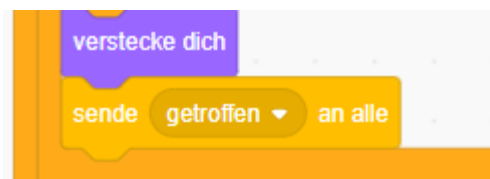


Soweit, so gut. Probiere es einmal aus und vergiss das Speichern nicht!

Schritt 8:

Alle Weltraumschrott-Figuren sollen nun dasselbe Programm erhalten, denn wenn die Kugel sie trifft, sollen sie alle verschwinden. Um dir Zeit und Mühe zu sparen, kannst du dieses eine Programm jetzt an für alle Figuren kopieren.

Vorher solltest du das Programm aber noch um eine Komponente erweitern. Nämlich „sende Nachricht „getroffen“ an alle“.



Aber jetzt zum Kopieren des Codes:

- Zieh mit gedrückter linker Maustaste, den Code auf eine weitere Figur
- Wenn die Figur zu wackeln anfängt, lass die Maustaste los.
- Jetzt wird das Programm zur Figur kopiert

- Klick die Figur an und schau nach, ob das Programm wirklich dort komplett angekommen ist.

Speichere das Spiel ab und probiere es einmal aus. Klappt alles?

Was fällt dir auf?

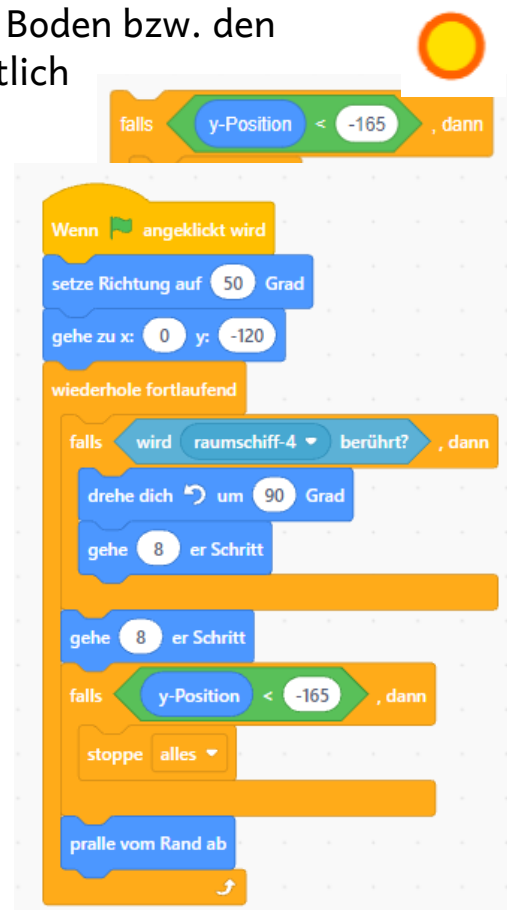
Selbst wenn du alle Objekte aufgeräumt hast, geht das Spiel weiter und es müsste auch noch etwas interessanter werden – also sollten wir noch eine „Gefahr“ einbauen. Wie wäre es, wenn das Spiel vorzeitig endet, weil die Kugel den unteren Rand berührt?

Klingt interessant, aber wie macht man das?

Schritt 9:

Wie findet man heraus, ob die Kugel den Boden bzw. den unteren Rand berührt hat? Das ist eigentlich ganz einfach.

- Wenn die Kugel einen bestimmten Y-Wert erreicht, hat bzw. wenn der Y-Wert kleiner als dieser Wert ist, ist der „Boden“ erreicht und das Spiel stoppt.
- In diesem Fall ist der Y-Wert -165
- Wenn also der Y-Wert kleiner als -165 ist, stoppt das Spiel.
- Diese Bedingung muss jetzt in das Programm der Kugel eingefügt werden.
- Schiebe sie hinter die erste Bedingung im Programm, so dass beide Bedingungen in der Schleife stecken.
- Damit die Kugel aber auch wirklich vom Rand abprallt, müssen wir jetzt noch den Befehl gehe 8er Schritt zwischen die zwei Bedingungen schieben.



Speichere dein Spiel!

Schritt 10:

Jetzt funktioniert das Spiel schon ganz gut. Du kannst den Weltraumschrott abschießen bzw. aufräumen und wenn die Kugel den unteren Rand berührt, stoppt das Spiel.



Aber es geht noch besser!

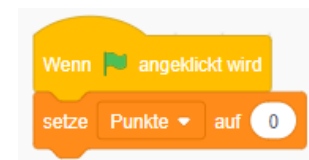
Wie wäre es mit Punkte zählen oder Geräusche machen oder ...

Außerdem muss das Spiel noch stoppen, wenn der komplette Weltraumschrott beseitigt wurde.

Es gibt also noch ein paar Baustellen, an denen wir weiterarbeiten können.

Fangen wir mit den Punkten an. Jedes Mal, wenn ein Stück Schrott getroffen wurde, soll ein Punkt dazugezählt werden. Dazu braucht es eine Variable, namens Punkte.

- Klicke auf das Bühnenbild ganz rechts unter Bühne. Jetzt wird das leere Programmfenster der Bühne angezeigt.
- Erstelle eine neue Variable und nenne sie Punkte. Wenn das Spiel beginnt, soll der Punktestand auf 0 gesetzt werden.
- Im nächsten Schritt soll folgendes passieren. Wenn die Nachricht „getroffen“ empfangen wird, soll der Punktestand um 1 erhöht werden.
- Die Variable „Punkte“ wird am linken oberen Bildrand angezeigt, so dass du den Punktestand beim Spielen verfolgen kannst.



Schritt 11:

Wann ist das Spiel zu Ende? Richtig! Wenn es keinen Weltraumschrott gibt – also alles aufgeräumt ist – ist das Spiel zu Ende oder wenn die Kugel den Boden berührt.

Aber gehen wir mal davon aus, dass die Kugel den Boden nicht berührt, dann sollte das Spiel nach 15 Punkten (es gibt 15 Teile Weltraumschrott) beendet bzw. gewonnen sein.

Aber wie sage ich das jetzt dem Programm?

- Es muss ein Vergleich her. Wenn das Spiel die 15. Nachricht „getroffen“ erhält, ist Schluss.
- Diese Abfrage wird in den Programmblock „Wenn Nachricht „getroffen“ empfangen“ integriert.
- Ok, jetzt stoppt das Spiel abrupt, wenn das 15. Teil Weltraumschrott getroffen wurde. Wahrlich ein unwürdiges Ende, oder?



Schritt 12:

Damit das Spiel nicht so plötzlich aufhört, wäre es doch schön einen „Gewinnersound“ abzuspielen und vielleicht die Nachricht, dass man das Spiel gewonnen hat. Generell ist Musik für ein Computerspiel eine gute Sache. Aber fang erst einmal langsam an.

Jedes Mal, wenn ein Stück Weltraumschrott verschwindet, soll ein Ton abgespielt werden. Wie funktioniert das? Eigentlich ist es gar keine große Sache.

- Sobald die Nachricht „getroffen“ empfangen wurde, ...
- ... wird dieser Sound abgespielt. Du kannst natürlich auch einen anderen Sound wählen.
- Danach werden die Punkte geändert

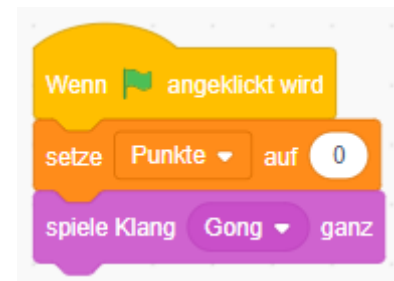


Schritt 13:

Wie wäre es mit einer Begrüßungston? Es könnte ein Gong schlagen, sobald das Spiel gestartet wird.



- Suche dir in den Klängen den Ton „Gong“ heraus und füge ihn zu deiner Klangsammlung
- Ändere den Code im Skript des Hintergrundes wie folgt:
- Füge „spiele Klang „Gong“ ganz“ ab zu den zwei Codeblöcken für den Start hinzu
- Speichere dein Spiel ab



Schritt 14:

Da wir schon bei den Klängen sind, wie wäre es mit einem Gewinnersound und einem Verlierersound?

Hört sich gut an. Also, gehen wir es an.

Der fröhliche Gewinnersound heißt „Win“ und du findest ihn in der Bibliothek Klänge

Der traurige Verlierersound heißt „Lose“ und du findest ihn ebenfalls in der Bibliothek Klänge.

Solltest du aber einen anderen Sound wollen, ist das auch kein Problem.

Wie also bekommen wir das Spiel jetzt dazu am Ende einen Gewinner- oder einen Verlierersound zu spielen?

Der Gewinnersound ist auf den ersten Blick kein Problem, denn es gibt ja bereits einen Programmblock auf der Hintergrundbühne, der feststellt, wenn alle 15 Objekte verschwunden sind.

Man könnte also, bevor der Befehl „Stoppe alles“ erteilt wird, den Block „spiele Klang „win“ ganz“ einfügen.



Das sieht soweit ganz gut aus, ABER es gibt noch ein Problem. Die Kugel fliegt weiter, während der Siegersound tönt. Theoretisch kann man jetzt tatsächlich noch verlieren.

Wie löst man das?

Indem eine Nachricht „gewonnen“ an alle geschickt wird, wenn die 15 Punkte erreicht wurden. Den Block baut man vor den Klang ein.



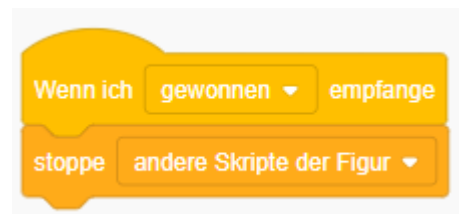
Schritt 15:

Bei Erreichen der 15 Punkte ist der Weltraumschrott aufgeräumt und nur die Kugel ist noch unterwegs. Deshalb ist die Kugel auch die Empfängerin der Nachricht.

Mit anderen Worten:

Zum Skript der Kugel kommt noch ein weiteres Skript hinzu

- Wenn die Nachricht „gewonnen“ empfangen wird
- Werden alle Skripte dieser Figur gestoppt



Schritt 16:

Wie ist es aber mit dem Verlierersound?

Im Skript der Kugel setzt du den Block „spiele Klang „lose“ ganz“ direkt über den Befehl „stoppe alles“.

Erstelle außerdem noch eine neue Nachricht „verloren“ und füge den Block „sende Nachricht „verloren“ an alle“ vor den „Klang-Block ein.



Schritt 17:

Warum wird noch die Nachricht verloren gebraucht?

Ganz einfach!

Mit den Nachrichten „gewonnen“ und „verloren“ kann man noch schöne Schilder einblenden, auf denen „gewonnen“ oder „verloren“ steht.

Das Schild „Gewonnen“

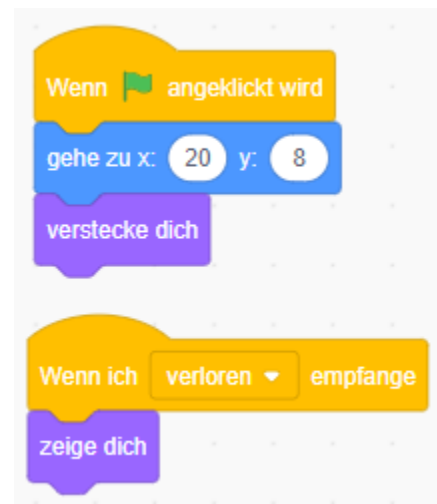
- Gehe zu den Figuren und klicke auf den Pinsel. Du gelangst jetzt zum „Kostümdesigner“
- Möchtest du ein eckiges Schild, ziehe ein Rechteck auf. Möchtest du ein rundes oder ovales Schild, wähle den Kreis.
- Wähle eine Füllfarbe und eine Randfarbe nach deinem Geschmack.
- Ziehe die gewünschte Form auf
- Wähle das Schreibtool und schreibe GEWONNEN.
- Beachte, dass du für die Schrift eine andere Farbe als für die Form wählst – sonst sieht man nichts 😊
- Gib der Figur den Namen „gewonnen“
- Wechsle in das Skript der Figur
 - Sobald das Spiel startet, soll sich das Schild in die Mitte der Bühne begeben und sich unsichtbar machen.
 - Sobald es die Nachricht „gewonnen“ erhält, soll es sichtbar werden.



Schritt 18:

Das „Verloren“-Schild funktioniert im Prinzip genauso

- Erstelle eine Form nach Wunsch
- Schreibe „VERLOREN“ hinein
- Gib der neuen Figur den Namen „verloren“
- Wechsle zum Skript der Figur
 - Sobald das Spiel startet, soll das Schild in die Mitte der Bühne gehen
 - Und unsichtbar werden
 - Sobald die Figur die Nachricht verloren erhält soll sie sichtbar werden



Schritt 19:

Teste dein Spiel!

Starte dein Spiel und spiele einmal drei Runden.

Beobachte dabei besonders die Kugel, wenn sie das Raumschiff trifft.

Vielleicht passiert es manchmal, dass die Kugel am Raumschiff hängen bleibt oder sich darin verheddert. Das liegt an der Form des Raumschiffs.

Wie könnten wir dafür sorgen, dass Scratch die Berührung einfacher erkennen kann?

Schritt 20:

Wir bauen eine Hitbox

In vielen Spielen sieht eine Figur ganz anders aus als die Fläche, mit der das Spiel Berührungen erkennt. Dafür gibt es **Hitboxen**.

Eine Hitbox ist eine meist unsichtbare Fläche, mit der das Spiel prüft, ob eine Figur oder ein Gegenstand getroffen oder berührt wird.

Besonders bei Figuren mit einer unregelmäßigen Form kann eine einfache Hitbox Berührungen zuverlässiger erkennen.

Auch unser Raumschiff bekommt jetzt eine Hitbox.

Eine Hitbox erstellen

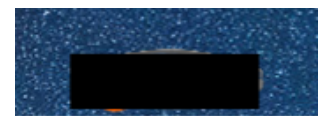
- Lege eine neue Figur an, indem du bei „Figur wählen“ auf den Pinsel (MALEN) klickst
- Nenne diese Figur im Kostüm-Editor „Hitbox“
- Zeichne mit dem Rechteck-Werkzeug ein ausgefülltes Rechteck, das in etwa so groß ist, wie das Raumschiff.
 - Erstelle das Rechteck über dem Mittelpunkt des Kostüm-Editors und NICHT am unteren Rand



Schritt 21

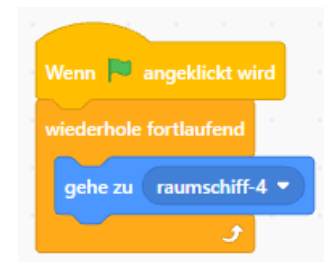
Die Hitbox programmieren

Die Hitbox soll immer beim Raumschiff bleiben und sich auch mit dem Raumschiff nach rechts und links bewegen.



Für die Hitbox brauchen wir folgenden Code

Die Hitbox soll sich immer zusammen mit dem Raumschiff bewegen. Deshalb lassen wir sie fortlaufend zur Position des Raumschiffs gehen.



Schritt 22

Für die Kugel müssen wir den bisherigen Code ändern:



Die Hitbox kann jetzt dem Raumschiff folgen. Aber unsere Kugel weiß noch nichts davon. Bisher prüft die Kugel, ob sie das Raumschiff berührt. Jetzt soll sie stattdessen auf die Hitbox reagieren. Deshalb müssen wir auch das Programm der Kugel anpassen. **Wähle die Figur „Kugel“ aus.**

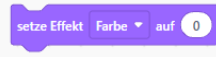
- Suche in ihrem Programm den Block, mit dem die Kugel prüft, ob sie Raumschiff 4 berührt. Ändere dort Raumschiff 4 in Hitbox.
- Damit sich die Kugel nicht in der Hitbox verfängt, lassen wir sie nach dem Abprallen sofort ein Stück weiterfliegen.

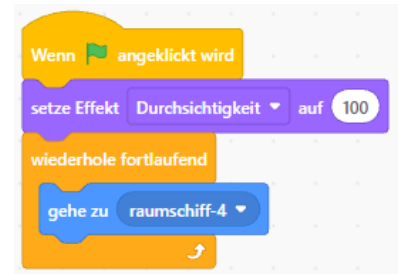


Schritt 23

Zu guter Letzt müssen wir die Hitbox noch unsichtbar machen, damit sie niemand sehen kann.



- Wechsle zur Figur „Hitbox“
- Füge den Block  nach dem Startblock ein und wähle statt Farbe Durchsichtigkeit. Setze den Wert auf 100 – das bedeutet totale Durchsichtigkeit. Niemand kann die Hitbox jetzt sehen.



Tipps & Tricks

- **Die Kugel ist dir zu schnell?**
Ändere die Zahl bei gehe (10)er Schritt. Eine kleinere Zahl macht die Kugel langsamer, eine größere schneller. Denk daran: Du hast den Block an **zwei Stellen** im Kugelprogramm.
- **Zu leicht oder zu schwer?**
Verteile den Weltraumschrott anders auf der Bühne. Je nachdem, wo die Teile liegen, verändert sich das Spiel. Lass zwischen den Schrottteilen aber etwas Platz, damit die Treffer zuverlässig gezählt werden.
- **Gestalte dein eigenes Spiel**
Du kannst andere Figuren als Weltraumschrott verwenden, ihre Größe verändern oder einen anderen Weltraum-Hintergrund wählen.
- **Probiere verschiedene Start-Richtungen aus**
Ändere die Zahl bei setze Richtung auf (50) Grad. Was passiert mit der Flugbahn der Kugel?

Materialien und Programmcode

Du findest alle Dateien für das Spiel auf folgender Website:

<https://cyber-kids.de/2024/10/22/wir-programmieren-ein-breakout-spiel/>

Damit du nicht alle Dateien in Scratch hineinladen musst, steht auch eine Datei als Grundgerüst zur Verfügung.

<https://scratch.mit.edu/projects/1386511288>