Das Arbeiten zu Hause mit minimalen Mitteln bietet die Chance, neue Inhalte Step-By-Step von Grund auf zu erkunden. Mit einem einfachen Mikrocontroller können, ohne Bauen und Basteln, spannende Experimente rund um die Robotik gemacht werden. Dies bietet viele Anknüpfungspunkte für spätere Schulprojekte im Makerspace, TTG oder Werken.

Das vorgestellte Projekt kann mit dem PGLU-Mikrocontroller KOMFORT, PIXEL oder jedem Arduino NANO/UNO umgesetzt werden. Details zum grafischen Programmieren des NANO/UNO gibt es hier.

1. Pong - ein altes Videospiel neu entdeckt

Das 1972 von Atari veröffentlichte «Pong» wurde zum ersten weltweit beliebten Videospiel und in den 1970er-Jahren zunächst auf Geräten in Spielhallen bekannt. Es gilt als Urvater der Videospiele und lässt sich mit einem einfachen Neopixel Strip leicht nachbauen und programmieren.

Das Spielprinzip von Pong ist simpel und ähnelt dem des Tischtennis: Ein Punkt (Ball) bewegt sich auf dem Bildschirm hin und her. Jeder der beiden Spieler steuert einen senkrechten Strich (Schläger), den er mit einem Drehknopf (Paddle) nach oben und unten verschieben kann. Lässt man den Ball am Schläger vorbei, erhält der Gegner einen Punkt (Quelle: Wikipedia).

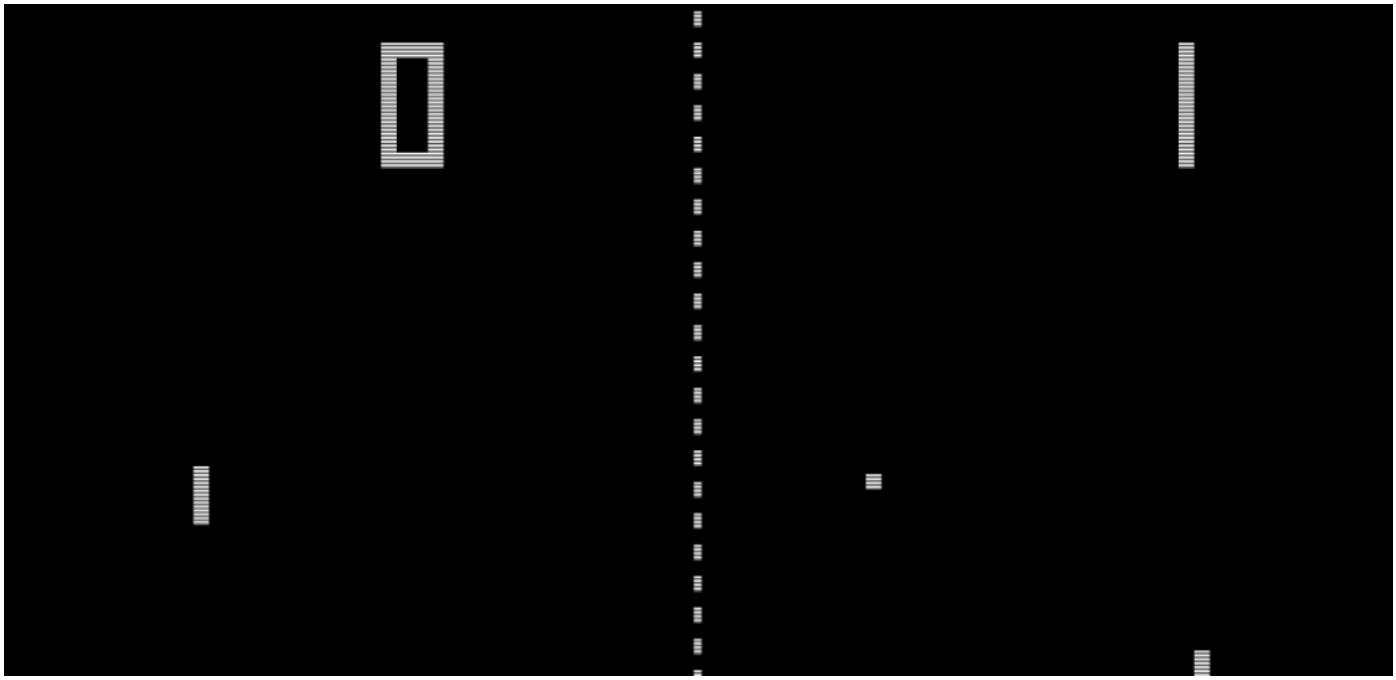


Bild 1: Bildschirmansicht "Pong". Quelle: de.pong-2.com

1.1. Videospiele - vom simplen Geschicklichkeitsspiel zum vielschichtigen Geschichtenerzähler

Als erstes Videospiel überhaupt, darf das Spiel «Tennis for two» angesehen werden. Es wurde 1958 in einem Forschungszentrum für Kernenergie im Bundesstaat New York als Nebenbeschäftigung entwickelt, jedoch nie verkauft. Dennoch war mit «Tennis for two» der Grundtyp aller Videospiele geboren: ein Bildschirm an den zwei Handcontroller angeschlossen sind (Bild 3).

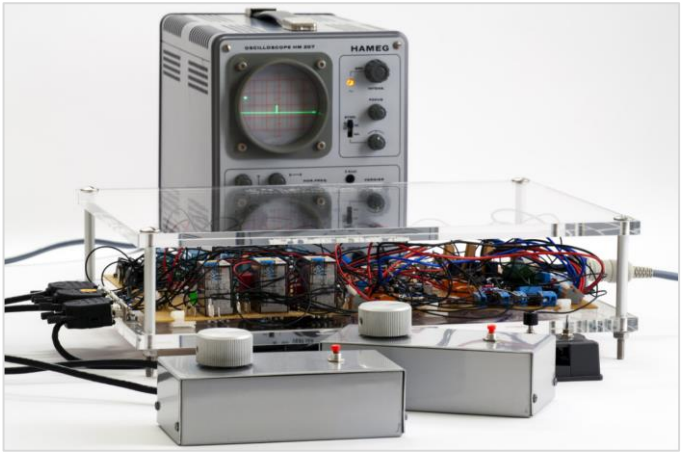


Bild 3 Tennis for two. Quelle: www.m-e-g-a.org

Bild 2 Spielkonsole "Pong" von Atari. Quelle: Wikipedia

Während die ersten Videospiele vor allem sportliche Geschicklichkeitsspiele waren, sind die heutigen Games komplexe virtuelle Welten, in denen sich Spieler*innen frei bewegen- und individuelle Abenteuer bestehen können. Dank der globalen Vernetzung über das Internet, entstehen Gamer Communitys in denen ganze Teams gegeneinander antreten.

Computerspiele sind damit zu einer der wichtigsten Freizeitaktivitäten für Jugendliche geworden und haben eine Weltweite Industrie entstehen lassen, welche eng mit der Filmindustrie verbunden ist.

Trotz dieser weitreichenden Entwicklung geht es auch beim Gaming der heutigen Zeit im Wesentlichen um die Fähigkeit, mit einem Steuergerät ein Ziel anzusteuern und durch Drücken eines Buttons treffen. Mit dem Arduino Mikrocontroller und einem Neopixel Streifen wollen wir genau dies tun und das Spiel «Pong» auf einfache Art selber bauen und programmieren.

2. Recherche

- Schaue das Video «Tennis fort wo» youtu.be/s2E9iSQfGdg
- Spiele «Pong» online unter de.pong-2.com/
- Schaue die Sendung «Medienkompetenz: Games» unter srf.ch/sendungen/myschool/games-2 (15 Min.) und finde heraus: welcher Spielertyp bist du?

3. Material

- Neopixel Stripe mit *71* Pixeln (letzten Pixel abschneiden, damit der Stripe einen Mittelpunkt hat)
- Mikrocontroller «Pixel» oder «Komfort»
- Zwei Schalter an den Sensoreingängen S1 und S2
- Zwei Potentiometer an den Sensoreingängen S3 und S4
- 1 Powerbank

4. Aufgabe - Pong bauen und programmieren

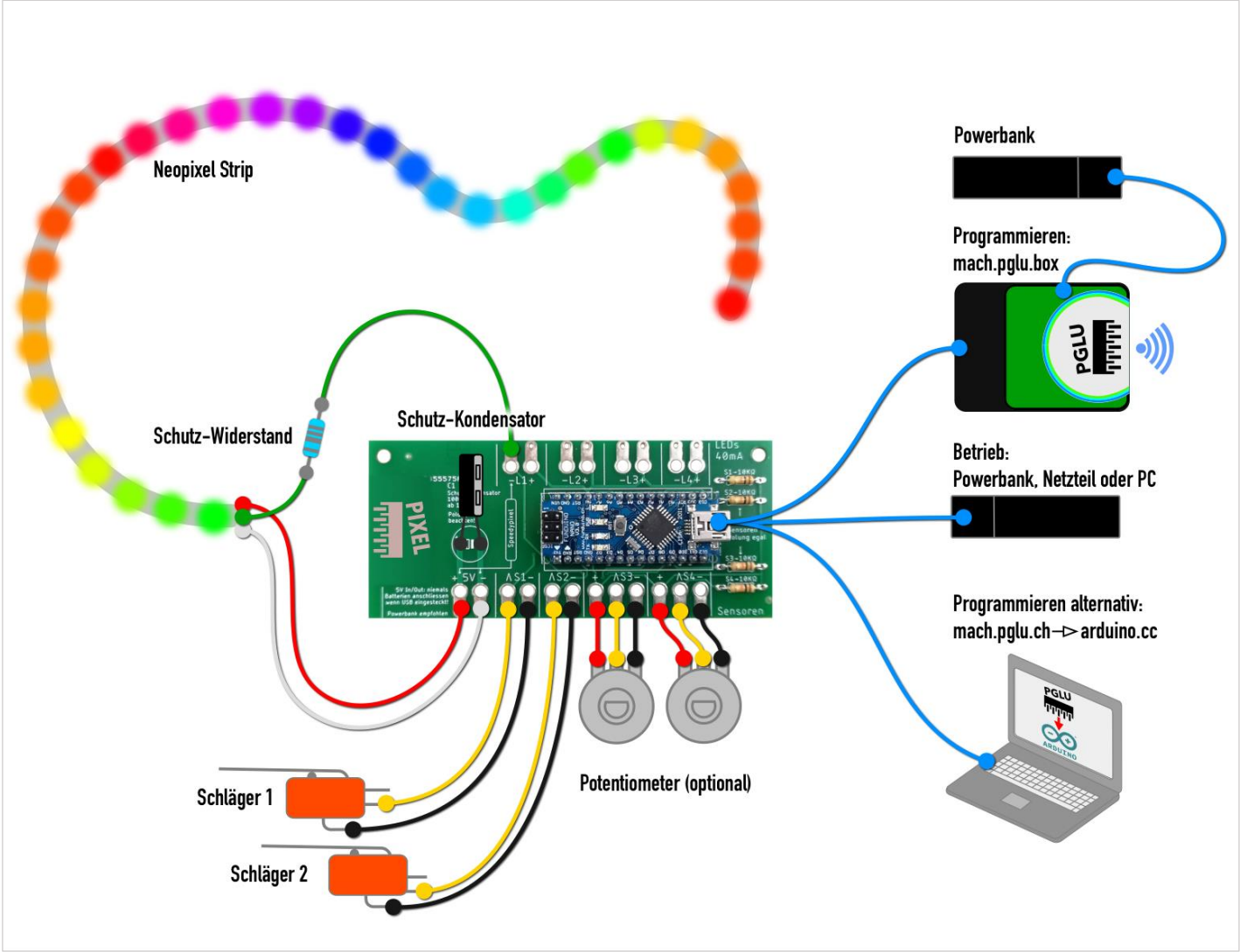
Baue und programmiere «Pong» nach der nachfolgenden Anleitung. Entwickle ein Gehäuse/Spieltisch für das Spiel welches durch sein Design eine eigene Geschichte rund um das Spiel erzählt. Erfinde eigene Erweiterungen und Zusatzfunktionen und gebe dem Spiel einen eigenen Namen.

4.1. Anschlussschema mit Mikrocontroller «Pixel»

Um Pong zu bauen brauchst du einen Mikrocontroller, einen Neopixel Stripe mit 71 Pixeln und einen oder zwei Schalter (single oder two-player Modus). Im Anschlussschema siehst du wie die Teile verbunden werden. Für das Verlöten der Komponenten folge diesen Videos:

> Schalter und Litzen löten: Video 1
> Mikrocontroller Pixel löten: Video 2

Die beiden Potentiometer brauchst du nur anzuschliessen, wenn du eine der Erweiterungen aus Kapitel 7 umsetzen möchtest.



4 Anschlussschema (Quelle: https://pglu.ch/was-sind-neopixel)

4.2. Aufbau des Programms in 4 Ebenen

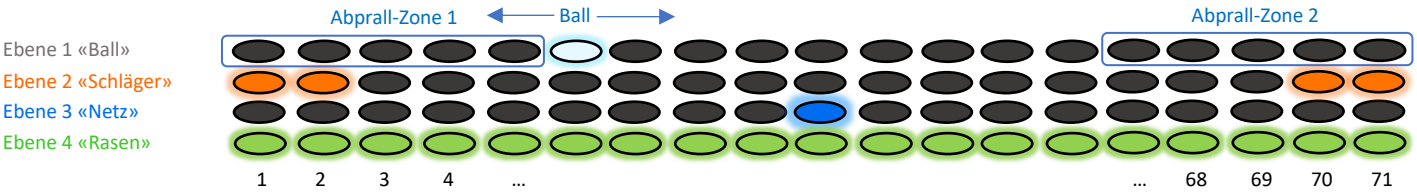
Um das Pong-Programm besser zu verstehen, stellen wir uns den Neopixel-Stripe als mehrere übereinander gelagerte Ebenen vor.

Als Betrachter siehst du immer das, was von oben her sichtbar ist, wenn du dir die dunkeln Pixel transparent vorstellst. Damit funktionieren Neopixel-Ebenen gleich wie Photoshop-Ebenen.

> **Ebene 1:** Der Ball läuft auf dem Neopixel-Stripe hin und her. An den Enden gibt es je eine Abprall-Zone
> **Ebene 2:** An den Enden des Neopixel-Stripes gibt es zwei Schläger

Optionen in Ebenen 3 und 4:

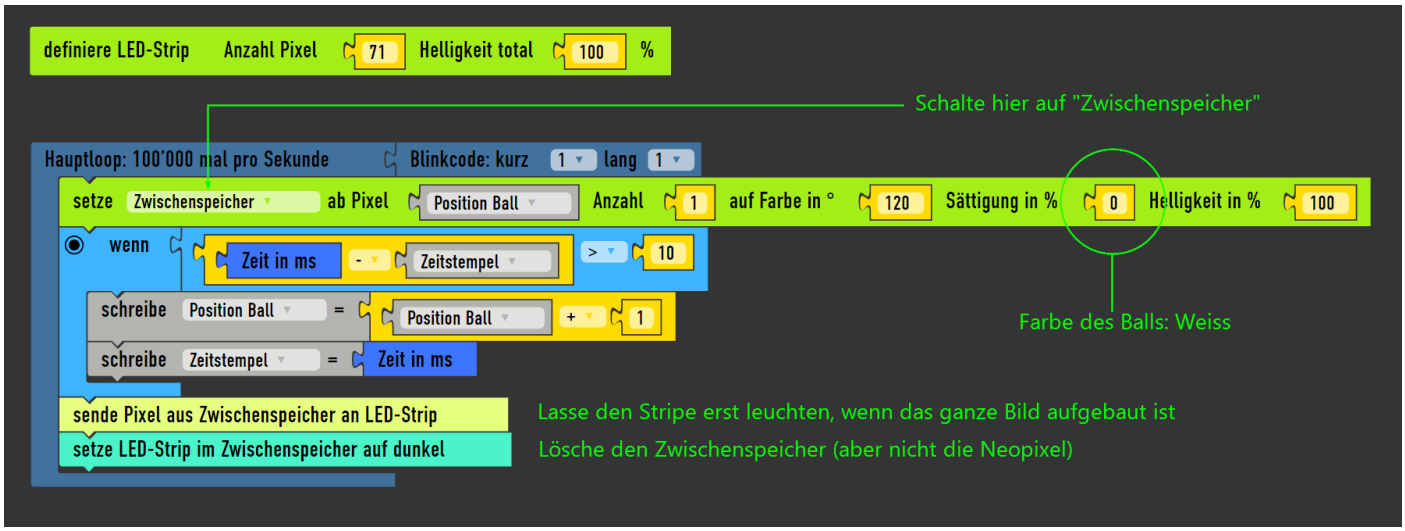
> **Ebene 3:** Der mittlere Pixel symbolisiert das Netz
> **Ebene 4:** Alle Pixel leuchten grün. Das ist der Rasen im Hintergrund



4.3. Ebene 1: den Ball auf dem LED-Stripe hochlaufen lassen

Da in «Pong» mehrere Dinge gleichzeitig ablaufen sollen, programmieren wir mit der Technik des Multitaskings. Lerne unter workshop.pglu.ch > Sketch > Zeit-in-ms, wie das mit Arduino funktioniert.

> Der Ball (ein Pixel) soll sich alle 10ms einen Pixel nach rechts verschieben. Dies machst du mit der einfachen Rechnung: Position=Position+1
> Später sollen *zeitlich parallel zu dieser Verschiebung* noch die beiden Taster (Schläger 1+2) abgefragt werden, ob sie gedrückt sind. Sobald dein Programm zwei Dinge gleichzeitig tun muss, spricht man von Multitasking. Das ist der Grund, weshalb dieser Programmteil auf den ersten Blick etwas kompliziert aussieht. Mehr dazu erfährst du unter workshop.pglu.ch > Sketch > Zeit-in-ms.



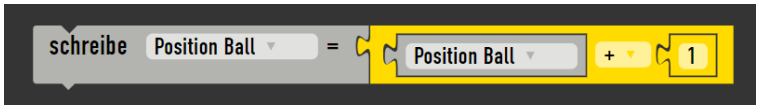
5 Den Ball Pixel für Pixel den Stripe hochlaufen lassen (Multitasking fähig)

4.4 Ebene 1: Der Ball wechselt die Laufrichtung, wenn eine der Abprall-Zonen erreicht wird

An beiden Enden des Stripes befinden sich Abprall-Zonen. Erreicht der Ball eine solche Zone, soll er automatisch seine Laufrichtung ändern:

> Abprall-Zone 1: Pixel 01 - 05
> Abprall-Zone 2: Pixel 67 - 71

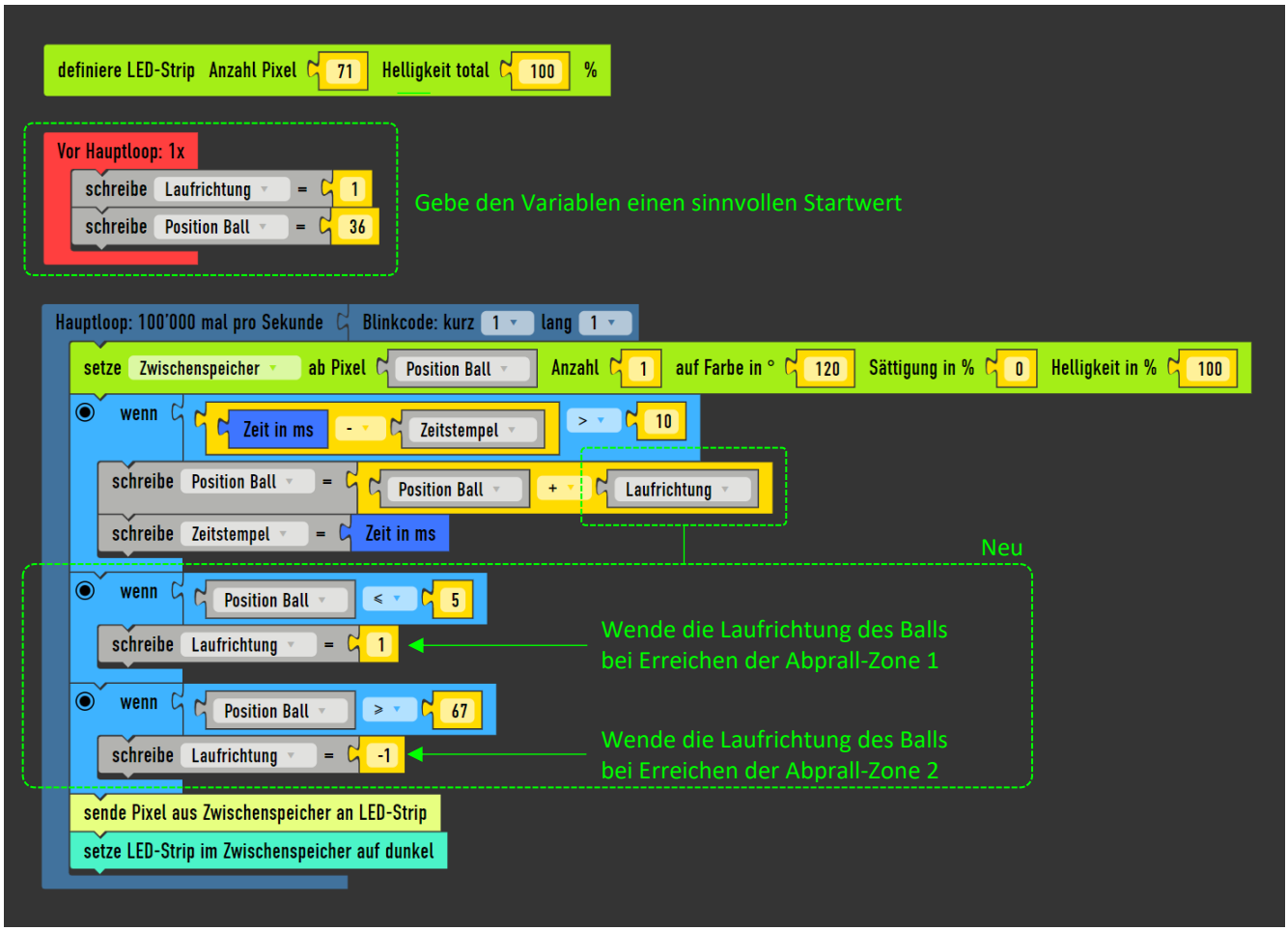
In Bild 5 geschieht das Hochlaufen des Pixels mit diesem Block:



Erweitere diesen Block so, dass die Position des Balls bei Erreichen der Abprall-Zone 2 um «1» vermindert wird. Dies programmierst du, indem das Vorzeichen der Zahl «1» vertauscht wird (+/-). Damit der Ball rückwärtslaufen kann, brauchst du also die Rechnung: Position=Position-1 anstatt Position=Position+1

Um die Laufrichtung genau an den Enden des LED-Stripes zu wenden, müssen folgende Fragen gestellt werden:

> Wenn Position kleiner als Beginn Abprall-Zone 1 (Pixel 5), dann wende die Laufrichtung
> Wenn Position grösser als Beginn Abprall-Zone 2 (Pixel 67), dann wende die Laufrichtung

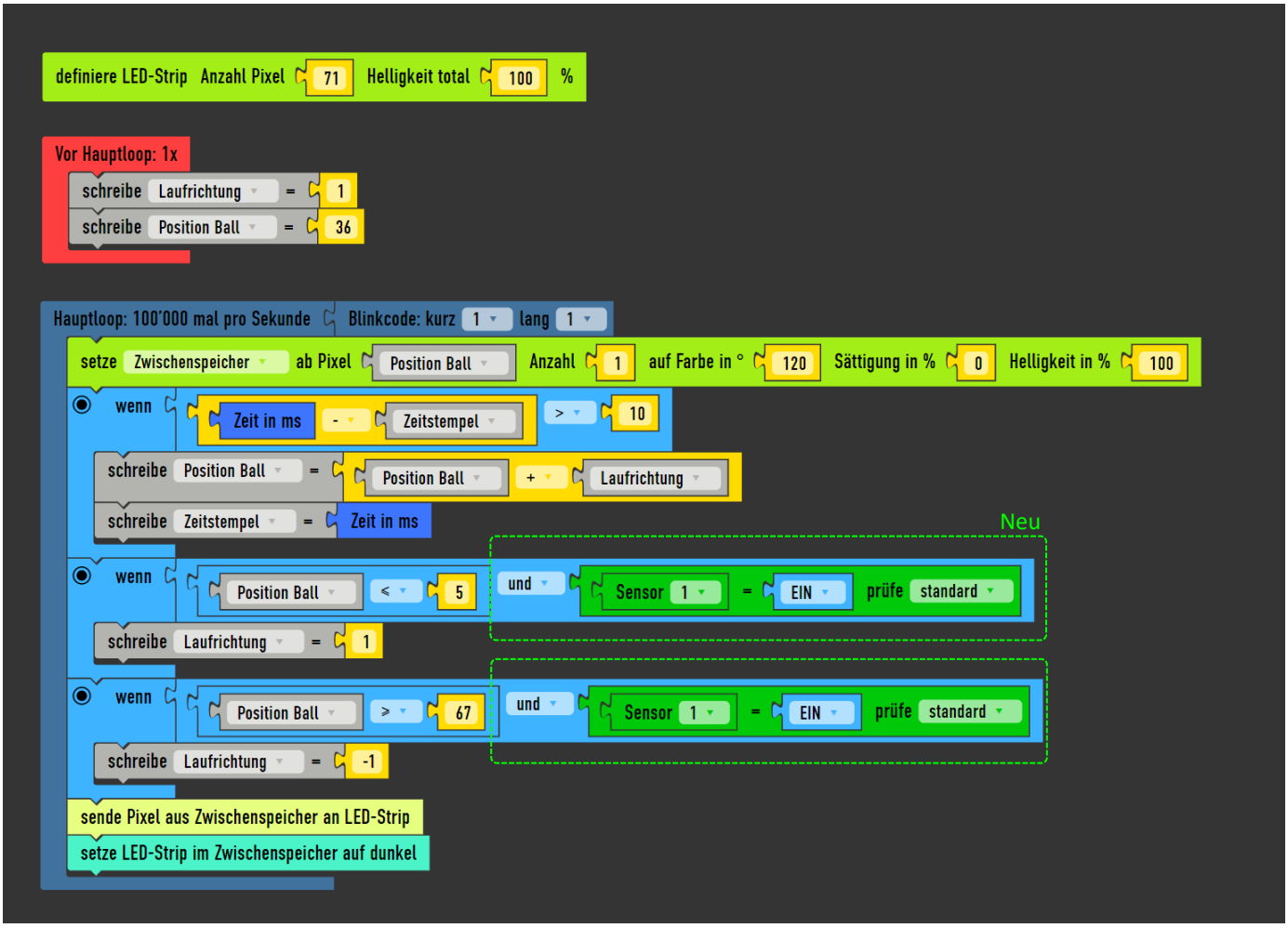


6 Programm für automatisches Pingpong

4.5 Ebene 1: Die Laufrichtung wechseln, wenn die Abprall-Zonen erreicht sind und der Button gedrückt ist

Bis jetzt hast du die Laufrichtung des Balls dann geändert, wenn die Abprall-Zonen erreicht wurden. Damit das Programm aber zu einem echten Geschicklichkeits-Spiel wird, bauen wir noch eine zweite Bedingung ein, die zum Wenden des Balls erforderlich ist:

1. Der Ball muss sich in der Abprall-Zone befinden (das haben wir ja schon programmiert)
   und
2. Der Button an Sensor 1 muss gedrückt sein



7 Grundprogramm "Pong" mit 1 Button für single player Modus

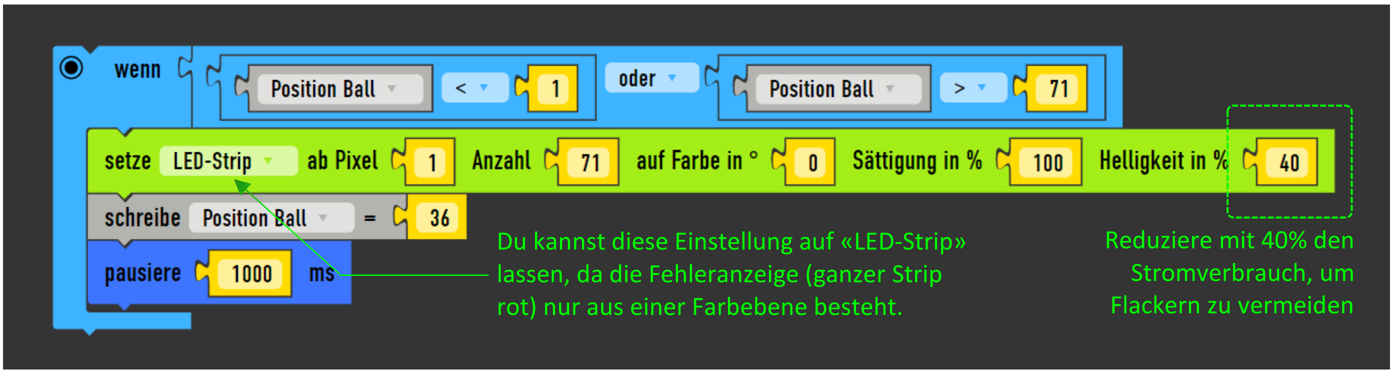
Dies ist das Grundprogramm, welches den Tennisball hin- und herfliegen lässt, sobald der Button gedrückt wird. Kannst du dir aber vorstellen, was geschieht, wenn der Button *nicht* gedrückt wird und der Ball ins Out geht?

In diesem Fall läuft dein Programm nämlich ganz normal weiter, ausser dass du keinen Ball mehr siehst. Dieser befindet sich eben im Out, was bedeutet, dass die Variable «Position Ball» ins unendliche hochgezählt wird, bis du den Button drückst. Teste das!

5. Zusatzfunktion Ebene 1: Fehleranzeige und automatischer Neustart

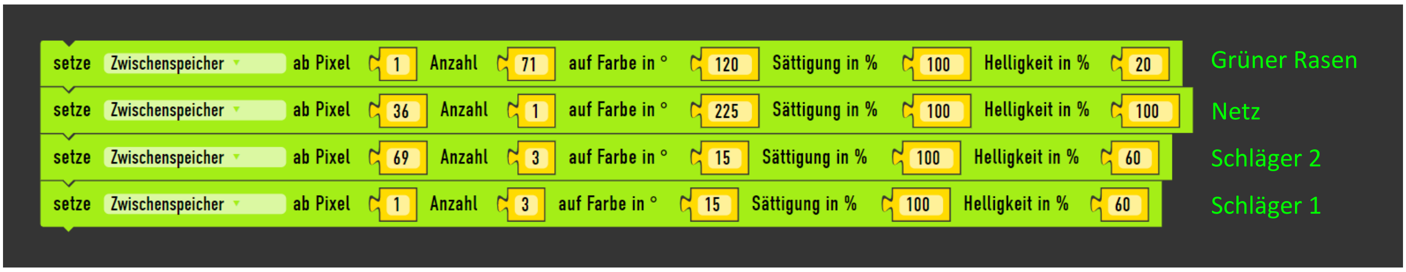
Dein Spiel ist nun fertig programmiert aber noch nicht perfekt. Was geschieht, wenn du beim Spielen einen Fehler machst und den Ball verpasst?

Füge darum diese Frage ein. Sie lässt den ganzen Stripe für 1 Sekunde rot leuchten (Fehler), sobald der Ball den Tennisplatz verlässt und versetzt den Ball wieder in die Ausgangsposition in der Mitte des Spielfeldes!



5.1. Ebenen 2-4, das Hintergrundbild: Grüner Rasen, orange Schläger und hellblaues Netz

Füge deinem Spiel einen grünen Hintergrund (Rasen/Pingpongtisch) und ein Netz in der Mitte hinzu. Setze dafür diese Blöcke zuoberst in deinen Hauptloop ein!



6. «Pong-1D Deluxe» als Video auf Vimeo anschauen

Kopiere diesen Code und füge ihn in den PGLU-Editor > Code > XML ein:


```
<xml xmlns="https://developers.google.com/blockly/xml"><variables><variable id="Nyx~tLu]@t~H42-v^p1j">Laufrichtung</variable><variable id="I!+NMK?ofEAWN831p;2.">Position Ball</variable><variable id="#w^AAFD`(XyXa||sVgQG">Tempo</variable><variable id="WJF-UkqG_R)=Vs%sqT12">Steigerung Tempo</variable><variable id="?jDc4tM#j~KO3V:;K|?w">Zeitstempel</variable></variables><block
type="set_neo_pixel" id="%_hC(e2Arbzc^+}EqL60" x="-228" y="-626"><value name="NUM_LEDS"><block type="math_number" id=":FIBB-JGeo$kT1N%@s.j"><field name="NUM">71</field></block></value><value name="BRIGHTNESS"><block type="math_number" id="@40yF}uPVw(|@LFCw,FP"><field name="NUM">100</field></block></value></block><block type="expert_setup"
id=")AxyrvoV_bM`9u=uXnT*" x="-228" y="-560"><statement name="CODE"><block type="variables_set" id="}f%`RO$.#Alv/r{YEaA|"><field name="VAR" id="Nyx~tLu]@t~H42-v^p1j">Laufrichtung</field><value name="VALUE"><block type="math_number" id="`01y40YkqKLd:os%,]Ll"><field name="NUM">1</field></block></value><next><block type="variables_set" id="p}GWAsU`(9AAT8%903=|"><field
name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field><value name="VALUE"><block type="math_number" id="Gsrj5m/--Pp:uB|ZWx6|"><field name="NUM">36</field></block></value><next><block type="variables_set" id="qkIfP2c79?$RV%XE{eJ*"><field name="VAR" id="#w^AAFD`(XyXa||sVgQG">Tempo</field><value name="VALUE"><block type="math_number"
id="R`.F4mW^w#xuMMX+#C$u"><field name="NUM">10</field></block></value></block></next></block></next></block></statement></block><block type="basic_main_loop" id="*5tQj3B57]iX,tU@,Wsf" deletable="false" x="-228" y="-391"><value name="BLINK"><block type="basic_blinkcode" id="dU+,V`P}v)HSFvpy50xs"><field name="SHORT">1</field><field
name="LONG">1</field></block></value><statement name="CODE"><block type="set_pixel_color" id="P3[`c@^.c.KiRXeRs+!Q"><field name="MODE">MODE2</field><value name="FROM"><block type="math_number" id="|7k*NzKFg0iuOPUMQtIm"><field name="NUM">1</field></block></value><value name="TO"><block type="math_number" id="az$`AuUWPAlYx7vnDk:Z"><field
name="NUM">71</field></block></value><value name="H"><block type="math_number" id="WY}7,Xm9ZM)DPQ*GSCa!"><field name="NUM">120</field></block></value><value name="S"><block type="math_number" id="z:f}4o;Q]XgA`N~1K,yo"><field name="NUM">100</field></block></value><value name="B"><block type="math_number" id="D-nfrPW;kJ%]tcY[)GV="><field
name="NUM">20</field></block></value><next><block type="set_pixel_color" id="X]e~k1Z}Sl{!_6Qc;e8n"><field name="MODE">MODE2</field><value name="FROM"><block type="math_number" id="pb{9ptS!#)6$z:bI$um="><field name="NUM">36</field></block></value><value name="TO"><block type="math_number" id="`^Xu3z3f(A(q[Os6mP:-"><field name="NUM">1</field></block></value><value
name="H"><block type="math_number" id="):}TJMx*35_vluIQ[7~?"><field name="NUM">225</field></block></value><value name="S"><block type="math_number" id="@D(Z0+f/}1eCVBy4{3=$"><field name="NUM">50</field></block></value><value name="B"><block type="math_number" id="U-@#Zb1t+w}e5fv4E7Cc"><field name="NUM">100</field></block></value><next><block type="set_pixel_color"
id="M]-$}YdWL^BY*jl3?eIL"><field name="MODE">MODE2</field><value name="FROM"><block type="math_number" id="{E;77-[UpQR9,u.u=%gn"><field name="NUM">69</field></block></value><value name="TO"><block type="math_number" id="- h]1@]SuVcb(uT#wk2Aw"><field name="NUM">3</field></block></value><value name="H"><block type="math_number" id="]OXOxhCyJ;T[uxB:{=@E"><field
name="NUM">15</field></block></value><value name="S"><block type="math_number" id="TCMT(,Bzy.6gkzIE}sQ]"><field name="NUM">100</field></block></value><value name="B"><block type="math_number" id="]}iQi$t[tib~6u.nJKH]"><field name="NUM">60</field></block></value><next><block type="set_pixel_color" id="x5BW.X.0t;0:~fjnU`k2"><field name="MODE">MODE2</field><value
name="FROM"><block type="math_number" id="t+P%a1#q;7y~@fEXKT_2"><field name="NUM">1</field></block></value><value name="TO"><block type="math_number" id="C`]T~Vkj/^b7~`t/{N$y"><field name="NUM">3</field></block></value><value name="H"><block type="math_number" id="C8E}`!aeIG[@s7r[;Rzn"><field name="NUM">15</field></block></value><value name="S"><block
type="math_number" id="5j|FRL8Gw/iHp@=5rfm;"><field name="NUM">100</field></block></value><value name="B"><block type="math_number" id="~Sb$5!7Q!r#ui(k)nZ%^"><field name="NUM">60</field></block></value><next><block type="set_pixel_color" id="L[!(/`#b3SPXf+z$Ka%."><field name="MODE">MODE2</field><value name="FROM"><block type="variables_get" id="-
nkRWMu[+/G0W|v|``Hm"><field name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field></block></value><value name="TO"><block type="math_number" id="=30s/i,kB:!gId!A-^lA"><field name="NUM">1</field></block></value><value name="H"><block type="math_number" id="Z0:]^k4[05t_7;^@jRPs"><field name="NUM">120</field></block></value><value name="S"><block type="math_number"
id="_s+?4)z5A^c]0SB{f,r1"><field name="NUM">0</field></block></value><value name="B"><block type="math_number" id="vcM]x_BPj8M#`D!F$s0c"><field name="NUM">100</field></block></value><next><block type="logic_if" id="JVc`SGM?6+sOTYwdl0rY"><value name="IF0"><block type="logic_compare" id="AM[b8:L)1O-YsFv|YYB!"><field name="OP">GT</field><value name="A"><block
type="math_arithmetic" id="d[Id-,*-+uR9lvjG2Mq."><field name="OP">MINUS</field><value name="A"><block type="flow_ms" id="j+n.mWu-5g35)^K*D#lp"></block></value><value name="B"><block type="variables_get" id="LF}bMZyZ{8E@q7qj}H*+"><field name="VAR" id="?jDc4tM#j~KO3V:;K|?w">Zeitstempel</field></block></value></block></value><value name="B"><block type="math_arithmetic"
id="A=pRr7K85+g5@#%*)J}0"><field name="OP">MINUS</field><value name="A"><block type="variables_get" id="dx#;B*:Ynom#I+q=cE~9"><field name="VAR" id="#w^AAFD`(XyXa||sVgQG">Tempo</field></block></value><value name="B"><block type="variables_get" id="9ct#b{YPnl:w;`^roRiv"><field name="VAR" id="WJF-UkqG_R)=Vs%sqT12">Steigerung
Tempo</field></block></value></block></value></block></value><statement name="DO0"><block type="variables_set" id="~?Ua5DStYtc|vlMe%G%#"><field name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field><value name="VALUE"><block type="math_arithmetic" id="zIpHL|+aPV,kx!#8WEPp"><field name="OP">ADD</field><value name="A"><block type="variables_get"
id="MN/a@s4cDs6!g[j.p`11"><field name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field></block></value><value name="B"><block type="variables_get" id="gV)21!lI^dtzhluZk^Nm"><field name="VAR" id="Nyx~tLu]@t~H42-v^p1j">Laufrichtung</field></block></value></block></value><next><block type="variables_set" id="*q%+.1!%k?y]l@Mn~^^!"><field name="VAR"
id="?jDc4tM#j~KO3V:;K|?w">Zeitstempel</field><value name="VALUE"><block type="flow_ms" id="-x-hRo?fM:T%h);L[-oZ"></block></value></block></next></block></statement><next><block type="logic_if" id=":/^=1=eb(h.1M#[4Zx#Q"><value name="IF0"><block type="logic_operation" id="7%BM$.zI{~{.k08|U8W}"><field name="OP">AND</field><value name="A"><block type="logic_compare" id="-
;g5Kk,QHSxK|7e#`)+J"><field name="OP">LTE</field><value name="A"><block type="variables_get" id="fT-m,zh|D#rf^GeCQkh_"><field name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field></block></value><value name="B"><block type="math_number" id="3!WyryJ^DOO^v9LEFmtT"><field name="NUM">5</field></block></value></block></value><value name="B"><block type="sensor_test_state"
id="(fwUJvcu!OrZ6=^Nd/Nq"><field name="MODE">STANDARD</field><value name="SENSOR"><block type="sensor_pin" id="7FYK^,H?@.2Z@?5:mBP~"><field name="SENSOR">SENSOR1</field></block></value><value name="STATE"><block type="basic_digital_state" id="=#s|ItF%8akZbHzM)gh8"><field name="STATE">LOW</field></block></value></block></value></block></value><statement
name="DO0"><block type="set_pixel_color" id="v`@j|Xz^tMnYjwDX9JD/"><field name="MODE">MODE2</field><value name="FROM"><block type="math_number" id="sXKw)KJ4ixFu6E[$BkG="><field name="NUM">1</field></block></value><value name="TO"><block type="math_number" id="V0gz@)Jm(qYIXa8=[;XD"><field name="NUM">7</field></block></value><value name="H"><block
type="math_number" id="~Hie,riYH,+DedlkTO)*"><field name="NUM">60</field></block></value><value name="S"><block type="math_number" id="QbkUgXgW(U!7RpVKt$h6"><field name="NUM">100</field></block></value><value name="B"><block type="math_number" id="FjVV:Xq++M`18u^f[4Dm"><field name="NUM">100</field></block></value><next><block type="fast_led"
id="_OlmECVs{r0$qS,`xumY"><next><block type="variables_set" id="oT|q2cSGp!fD9BbZ%vbp"><field name="VAR" id="Nyx~tLu]@t~H42-v^p1j">Laufrichtung</field><value name="VALUE"><block type="math_number" id="XwHmCwARVu@uhwm6a%:2"><field name="NUM">1</field></block></value><next><block type="variables_set" id="Y;YYJGJqk_w%/GAh(yBZ"><field name="VAR"
id="#w^AAFD`(XyXa||sVgQG">Tempo</field><value name="VALUE"><block type="math_number" id="5B(1uzIa`M)smz8EI1Ng"><field name="NUM">6</field></block></value><next><block type="flow_delay" id="TW?mu+|e^fx=g/fGZ~.S"><value name="DELAY"><block type="math_number" id="}fCa)[RvL%(X:2P9)j(o"><field
name="NUM">40</field></block></value></block></next></block></next></block></next></block></next></block></statement><next><block type="logic_if" id="YixEX;L$,RZE1P:7NzF}"><value name="IF0"><block type="logic_operation" id="75}+,enSaHs?mU{F0oRE"><field name="OP">AND</field><value name="A"><block type="logic_compare" id="6*KObp-9V$-7~0Mnc1vn"><field
name="OP">GTE</field><value name="A"><block type="variables_get" id="0)e?m7e,XRaWFg^6?5X6"><field name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field></block></value><value name="B"><block type="math_number" id="Y#oQdJsmmHVdWWn3+pHJ"><field name="NUM">67</field></block></value></block></value><value name="B"><block type="sensor_test_state"
id="kZ^~o3X6[$Lz$vy!VR7E"><field name="MODE">STANDARD</field><value name="SENSOR"><block type="sensor_pin" id="RL?iU~M^nc+K/RYR_yBI"><field name="SENSOR">SENSOR1</field></block></value><value name="STATE"><block type="basic_digital_state" id="At^:k1%jG;KZeLZzL/2-"><field name="STATE">LOW</field></block></value></block></value></block></value><statement
name="DO0"><block type="set_pixel_color" id="j=FLd#/;9[Cz@lH)d@-i"><field name="MODE">MODE2</field><value name="FROM"><block type="math_number" id="mV],LbLk;tT3;g+=c1t6"><field name="NUM">65</field></block></value><value name="TO"><block type="math_number" id="VF0p3rupEEU$~a+)):Rr"><field name="NUM">7</field></block></value><value name="H"><block
type="math_number" id="GKtYZ$$-yv,St6iK,qiw"><field name="NUM">60</field></block></value><value name="S"><block type="math_number" id="`Fq1(x6e)$KZ~RZKBqOA"><field name="NUM">100</field></block></value><value name="B"><block type="math_number" id="La#gSs:qvZDZ.9(Ca.H`"><field name="NUM">100</field></block></value><next><block type="fast_led"
id="Ch45Q@K3R8P}01I9hMS}"><next><block type="variables_set" id="mfOl%55EQ_B8Hi%LTCqn"><field name="VAR" id="Nyx~tLu]@t~H42-v^p1j">Laufrichtung</field><value name="VALUE"><block type="math_number" id="I-+PRz|oi8mvIaB}subs"><field name="NUM">-1</field></block></value><next><block type="variables_set" id="xaW?y!ycMCnK#6Z*8:[:"><field name="VAR"
id="#w^AAFD`(XyXa||sVgQG">Tempo</field><value name="VALUE"><block type="math_number" id="06i;bzB[6X;Ur}XT|XB-"><field name="NUM">6</field></block></value><next><block type="flow_delay" id="HCSv?^/LSZzIs4rhDr}t"><value name="DELAY"><block type="math_number" id="n4wpE44wNh_)$oT6vq68"><field
name="NUM">40</field></block></value></block></next></block></next></block></next></block></next></block></statement><next><block type="logic_if" id="EgAX2}]Srhp9`}a^r3]8"><value name="IF0"><block type="logic_operation" id="M*1U|iY}g+$e`~am}.aK"><field name="OP">OR</field><value name="A"><block type="logic_compare" id="U2T-C=zs)5b-J%k+[}I*"><field name="OP">LT</field><value
name="A"><block type="variables_get" id="|ha#?xRzeo!An%:_Y)IA"><field name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field></block></value><value name="B"><block type="math_number" id="[:ua*A^Gd02`3bq$hg|Q"><field name="NUM">1</field></block></value></block></value><value name="B"><block type="logic_compare" id=":r*wH,SiK+lewSPr!CRf"><field name="OP">GT</field><value
name="A"><block type="variables_get" id="5(N/fPDPf`$?VnE_ziYo"><field name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field></block></value><value name="B"><block type="math_number" id="vqT/)%$I4pePT*G#9VNu"><field name="NUM">71</field></block></value></block></value></block></value><statement name="DO0"><block type="procedures_callnoreturn"
id="nOh?4qx:~kaukYL`}{Fv"><mutation name="Fehleranzeige"></mutation></block></statement><next><block type="variables_set" id="L}n7XL$a[y`%0WOC*]:E"><field name="VAR" id="#w^AAFD`(XyXa||sVgQG">Tempo</field><value name="VALUE"><block type="math_arithmetic" id="U3LOD7rwk*H4w,R8clo+"><field name="OP">ADD</field><value name="A"><block type="variables_get"
id="%CT/cB[3t*FJR:c*:@IO"><field name="VAR" id="#w^AAFD`(XyXa||sVgQG">Tempo</field></block></value><value name="B"><block type="math_number" id="%m|4(cjL_MXhnjbu@;%I"><field name="NUM">0.1</field></block></value></block></value><next><block type="variables_set" id=",Z.[I8C[roh/GZ}v|NSR"><field name="VAR" id="WJF-UkqG_R)=Vs%sqT12">Steigerung Tempo</field><value
name="VALUE"><block type="math_arithmetic" id="LQfSh]ky~rw8_7BaoNU/"><field name="OP">ADD</field><value name="A"><block type="variables_get" id="ukhY`D|_8uhl4PCj.EOM"><field name="VAR" id="WJF-UkqG_R)=Vs%sqT12">Steigerung Tempo</field></block></value><value name="B"><block type="math_number" id=")4Y0Img5U#8XnVczClMP"><field
name="NUM">0.0003</field></block></value></block></value><next><block type="fast_led" id="Y0P_35lDnM{~IQO{9x#c"><next><block type="fast_led_clear"
id="*(V@[?dp}yt8b%Pj+b]{"></block></next></block></next></block></next></block></next></block></next></block></next></block></next></block></next></block></next></block></next></block></next></block></next></block></statement></block><block type="basic_comment" id="-1hSIHqcv9RoZVRc-]:~" x="662" y="-356"><field name="COMMENT">grüner Rasen</field></block><block
type="basic_comment" id="FJN:*8bJ+U!f^qT=c9;v" x="659" y="-316"><field name="COMMENT">Netz</field></block><block type="basic_comment" id="pklX)sRg+nea)hQ+ck{b" x="661" y="-273"><field name="COMMENT">Schläger 2</field></block><block type="basic_comment" id="[)DVWO7bCLh0o^T$:;Tj" x="662" y="-233"><field name="COMMENT">Schläger 1</field></block><block type="basic_comment"
id="H]lNZt@)9`lQVe??4%V*" x="730" y="-191"><field name="COMMENT">Ball</field></block><block type="basic_comment" id="Z!%M[EhhxL]*6_^Q[.C3" x="566" y="-144"><field name="COMMENT">Position des Balls im Verlauf der Zeit</field></block><block type="basic_comment" id="bmx@7*Y/;4.mhPeZUx:H" x="586" y="-8"><field name="COMMENT">Abfrage Schläger 1</field></block><block
type="basic_comment" id="I1UL2qD,ADNY43OeP!_t" x="121" y="82"><field name="COMMENT">Wende der Laufrichtung</field></block><block type="basic_comment" id="^8wmpYb.{];ZjyRSHI(O" x="589" y="233"><field name="COMMENT">Abfrage Schläger 2</field></block><block type="basic_comment" id="Ax;Av7;PWCEd.L.nrLh-" x="118" y="317"><field name="COMMENT">Wende der
Laufrichtung</field></block><block type="basic_comment" id="c]ZQsJ^n|OE04jA%|F$X" x="474" y="472"><field name="COMMENT">Fehler, falls Ball im Out</field></block><block type="basic_comment" id="vVl`:*_{AEHp|.C0,42B" x="-577" y="561"><field name="COMMENT">Lässt Ball in der Luft langsamer werden</field></block><block type="basic_comment" id="{W[pRiX^])QD4jIB|6Un" x="-579"
y="606"><field name="COMMENT">Lässt Spiel mit der Zeit schneller werden</field></block><block type="basic_comment" id="ZyeFI!CjE{]?0C]e_!gy" x="-228" y="763"><field name="COMMENT">Unterprogramm "Fehler"</field></block><block type="procedures_defnoreturn" id="gikvOX5?{#$zPAESQ;uN" x="-229" y="807"><field name="NAME">Fehleranzeige</field><statement name="STACK"><block
type="set_pixel_color" id="NLquNe2%NPTCHjQUv/04"><field name="MODE">MODE1</field><value name="FROM"><block type="math_number" id="DMp-=^6XT.vWc{arjNtJ"><field name="NUM">1</field></block></value><value name="TO"><block type="math_number" id="YWER2QfhxTyRM3adny%L"><field name="NUM">71</field></block></value><value name="H"><block type="math_number"
id="js3%sDF!Tj1T2/#G?y5s"><field name="NUM">0</field></block></value><value name="S"><block type="math_number" id="09I%tgE~Lw2qzi5_g^;)"><field name="NUM">100</field></block></value><value name="B"><block type="math_number" id=":.m+SmwAOn1[9cKp)F,3"><field name="NUM">40</field></block></value><next><block type="variables_set" id="*zZf^$CqcKX#)A/- 8]GA"><field name="VAR"
id="#w^AAFD`(XyXa||sVgQG">Tempo</field><value name="VALUE"><block type="math_number" id="_ujGBr}YfhMyE/IHVDOK"><field name="NUM">10</field></block></value><next><block type="variables_set" id="R1No+~{!pO_Eo(fjCQW8"><field name="VAR" id="I!+NMK?ofEAWN831p;2.">Position Ball</field><value name="VALUE"><block type="math_number" id="iR#x5yaYOXeOsv-j!Kd="><field
name="NUM">36</field></block></value><next><block type="variables_set" id="Rfy*?w5!dOhF[u~vp/W$"><field name="VAR" id="WJF-UkqG_R)=Vs%sqT12">Steigerung Tempo</field><value name="VALUE"><block type="math_number" id=",5rhp^-{^Xd!/*s:CDd["><field name="NUM">0</field></block></value><next><block type="flow_delay" id="J5:f1fmtAm5)rARLtA0%"><value name="DELAY"><block
type="math_number" id="h9:o!Fl#^nky=oL,)s|1"><field name="NUM">1000</field></block></value></block></next></block></next></block></next></block></next></block></statement></block></xml>
```


Auch diese Version hat noch Schwächen: Wird der Button ständig niedergehalten, kann nie ein Fehler passieren und der Ball fliegt ewig hin und her. Ändere das, indem du das Programm so erweiterst, dass ein Drücken des Buttons *ausserhalb der Abprall-Zone* zu einem Fehler führt!

7. Erweiterungen

Programmiere Erweiterungen, die das Spiel spannender machen:

- Beschleunigung des Spiels mit zunehmender Spieldauer
- Die Möglichkeit, den Schläger für einen Angriff (Volley) mit einem Potentiometer zum Netz zu verschieben
- Zweispieler Modus und Punktezähler
- Ein zweiter Ball, der nach einer bestimmten Zeit ins Spiel kommt
- Soundeffekte (siehe workshop.pglu.ch/Aktor/Piezo-Element)

8.1. Zeit

- 4-16 Lektionen (Je nach Bau und Design)

8.2. Erforderliche Komponenten für diese Aktivität

- Neopixel Stripe mit 72 Pixeln
- Mikrocontroller «Pixel» oder «Komfort»
- Zwei Schalter an S1 und S2
- Zwei Potentiometer an S3 und S4
- 1 Powerbank
- Mac oder PC mit USB-Anschluss

8.3. Mikrocontroller mit dem PC verbinden

- Video - Arbeiten zu Hause
- Video - Arbeiten in der Schule mit Lerngruppen

8.4. Erforderliche Komponenten für alle Aktivitäten der Serie

- Mikrocontroller «Komfort» oder Arduino NANO/UNO
- Sensor Schalter
- Sensor Potentiometer
- Sensor Mikrofon
- Sensor Ultraschallsensor
- Aktor Neopixel
- Jumper Kabel Stecker-Buchse
- PC oder Mac mit USB
- Schraubenzieher Grösse 0 oder 1
- Klebstreifen

9. Unterstützung

- info@pglu.ch

Impressum

- PGLU.CH
  Seminarstrasse 68
  5430 Wettingen
  https://pglu.ch
  https://workshop.pglu.ch

Bildnachweise

- Titelbild: pglu.ch
- Bild 1: !Original: Bumm13Vector: Beao (https://commons.wikimedia.org/wiki/File:Pong.svg), „Pong“, als gemeinfrei gekennzeichnet, Details auf Wikimedia Commons: https://commons.wikimedia.org/wiki/Template:PD-ineligible
- Bild 2: Evan-Amos (https://commons.wikimedia.org/wiki/File:TeleGames-Atari-Pong.png), „TeleGames-Atari-Pong“, https://creativecommons.org/licenses/by-sa/3.0/legalcode
- Bild 3: zvg, MEGA Museum of Electronic Games & Art e.V. https://www.m-e-g-a.org/
- Grafiken: pglu.ch