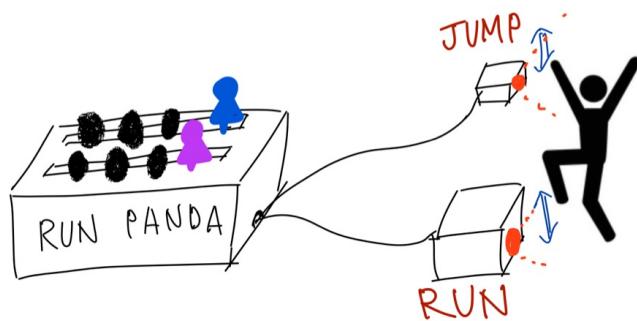
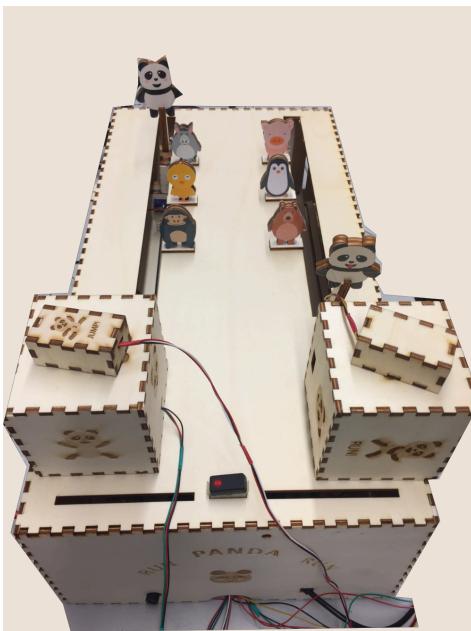


"RUN PANDA RUN

- ein interaktives Jump'n'Run Spiel mit Hinderniserkennung-Sensor"



Von der Idee zum Spielkonzept

Spielidee: Für das Thema "Ein interaktives Spielsystem (Design of interactive Systems)", mit dem sich die Spieler idealerweise viel bewegen können, hatten wir einige Ideen gesammelt. Aber keine der Ideen hatte uns besonders gefallen, da sie nicht so kreativ waren. Daher haben wir uns eine andere Frage gestellt: Können wir eine klassische Spielidee nehmen und um ein neues Konzept erweitern? Jump'N'Run ist eine weitverbreitete Art von Spielen, das normalerweise über einen Controller oder mit wenigen Tasten gesteuert wird. Die Spielfigur muss nur laufen und springen.

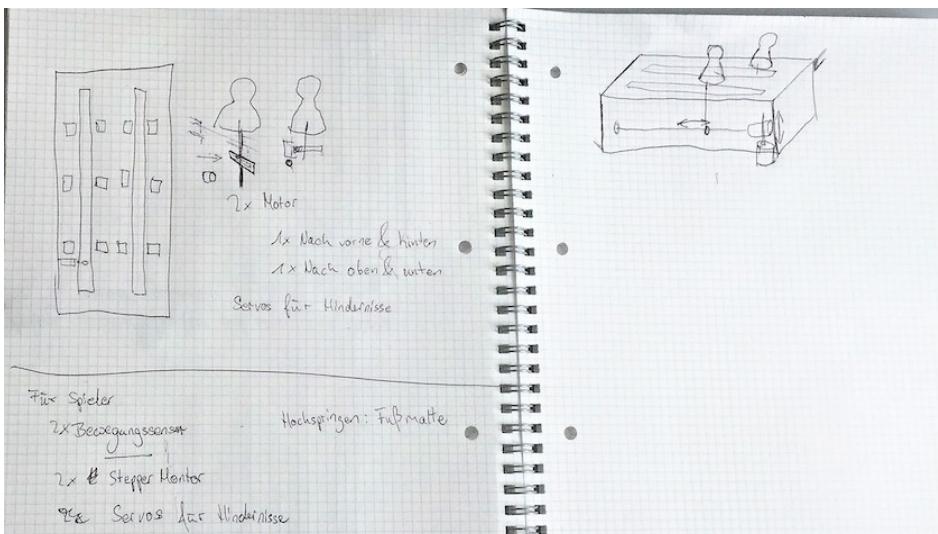
-> Um Spielfiguren zu steuern müssen sich die Spieler auch bewegen, entsprechend der Art der Bewegung. Also, schnell laufen, dann läuft die Figur auch! Springst du, dann springt deine auch deine Figur!

Konzept: Unser Spiel ist ein interaktives und dynamisches Jump'n'Run Spiel für zwei Personen, welches den Spieler auffordert sich selbst zu bewegen um so die Spielfigur zu bewegen. Die Spieler müssen sich vor Controllern (einer für Springen und einer für Rennen) bewegen während diese Bewegungen von Sensoren erkannt werden.

Run Panda Run

Zwei Panda konkurrieren miteinander um schneller ans Ziel zu kommen. Auf dem Weg zum Ziel stehen einige Tiere (Hindernisse) und Spielfiguren müssen sie überspringen. Wenn sie gegen ein Hindernis stoßen, werden sie um einige Schritte zurückgesetzt. Um das Spiel zu gewinnen, sollten sich die Spieler möglichst schnell und korrekt vor den Controllern bewegen.

Hardware



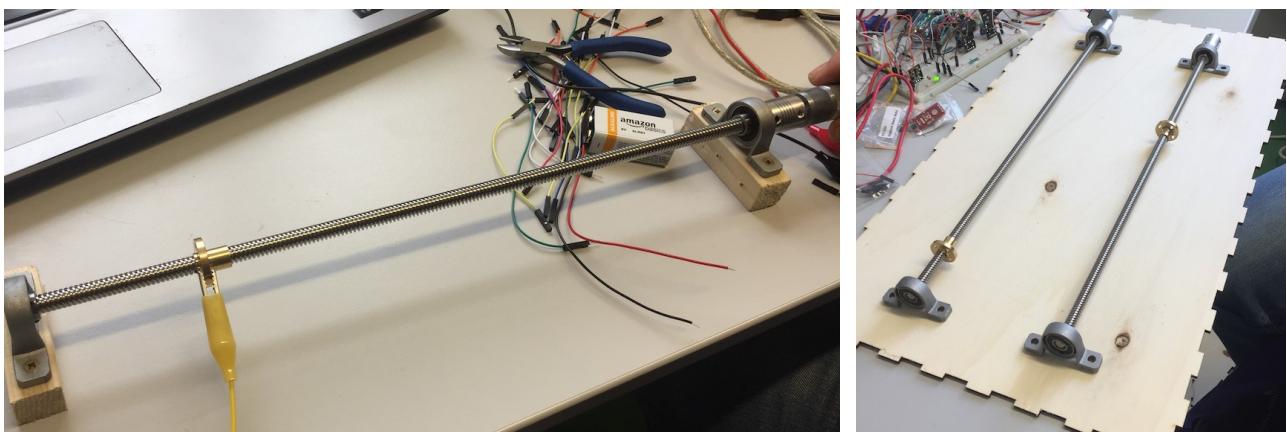
(Skizze des Spieldraufbaus)

Materialliste:

Arduino MEGA
Motorshield
2x Motoren
2x Servomotoren
Gewindespindel
4x Hindernissensoren
Schalter
LED
Widerstände 12 Ω, 150 Ω

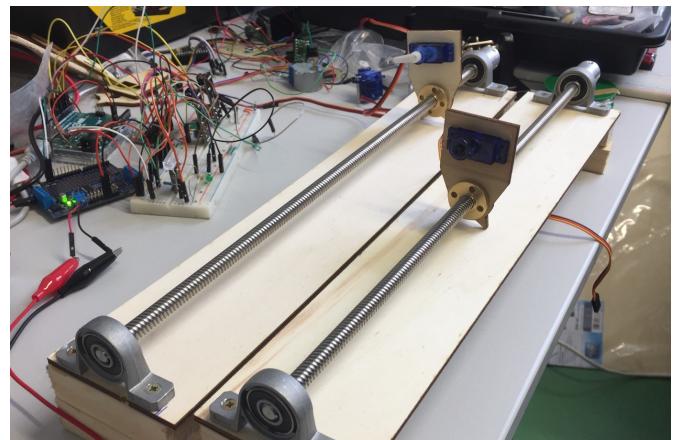
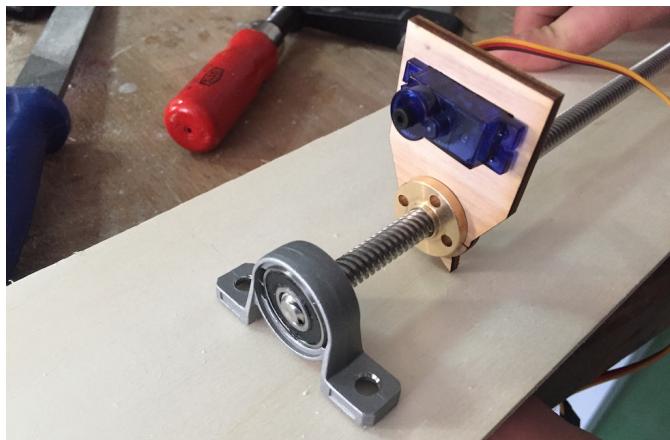
Aufbau der Rennstrecke zum Laufen

Um „RUN PANDA RUN“ umzusetzen haben wir an unseren Arduino MEGA ein Motorshield angeschlossen. An das Motorshield haben wir zwei Motoren gesetzt und eine externe Stromquelle, die uns 12 Volt liefert, damit die Motoren mit voller Leistung arbeiten können. Die Motoren haben wir an jeweils einer Gewindespindel fest geschraubt. Für das bewegliche Stück der Spindel haben wir mit dem Lasercutter eine Vorrichtung für die Spielfigur geschnitten.



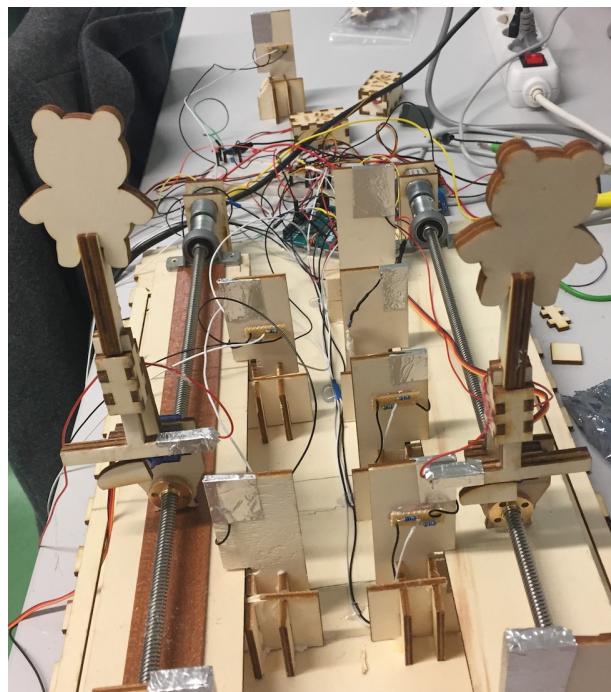
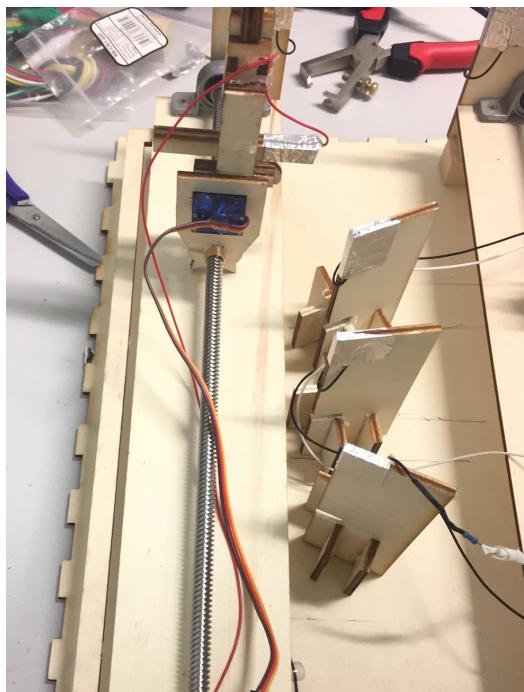
Springen mit Servo

In dieser Vorrichtung haben wir auch einen Servomotor verbaut, welcher die Figur nach oben bewegen kann, wodurch wir den Sprung über die Hindernisse realisieren konnten. Damit aber auch das Programm erkennen kann, dass ein Hindernis getroffen wurden, haben wir an den beweglichen Teil der Vorrichtung für die Spielfigur ein Aluklebeband geklebt, und dieses mit einem Kabel verbunden, welches ständig mit 5V verbunden ist.



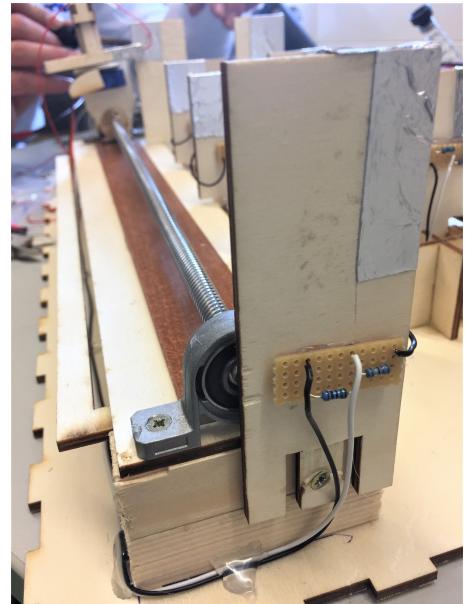
Aufnehmen der Hindernisse

An die Hindernisse haben wir ein anderes Stück Aluklebeband geklebt und dieses mit einem Kabel verbunden, welches zu zwei in Reihe geschaltete Widerstände führt. Dabei ist der Erste ein kleiner Widerstand, bei uns $12\ \Omega$, und der Zweite ein größerer Widerstand, bei uns mit $150\ \Omega$. Zwischen den beiden Widerständen geht ein Kabel weg, das mit einem Pin unseres Arduinos verbunden ist. Hinter den beiden Widerständen führt ein Kabel zum GND. Somit wird ein Stromkreis geschlossen, wenn das Aluklebeband der Figur mit dem Aluband des Hindernisses in Berührung kommt. Dabei haben wir darauf geachtet, dass wenn der Servomotor der Spielfigur oben ist das Hindernis nicht berührt wird. Falls ein Hindernis berührt wird fährt der Motor zurück und der Spieler wird zurück geworfen und muss das Hindernis noch einmal bewältigen.



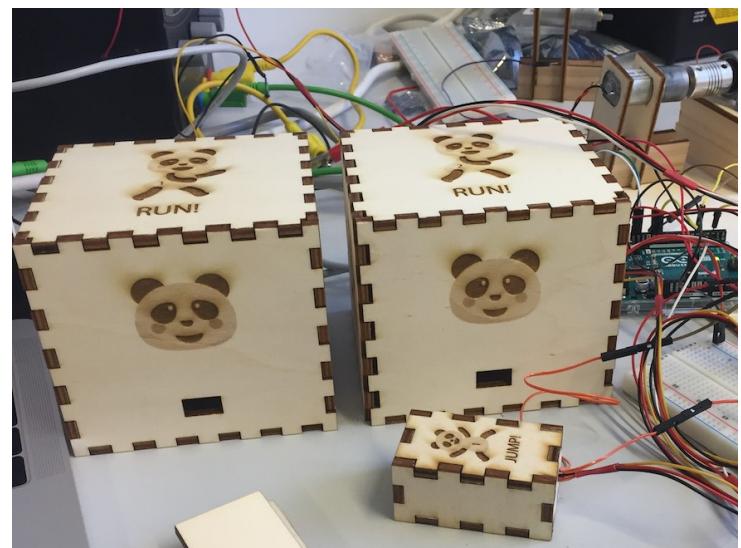
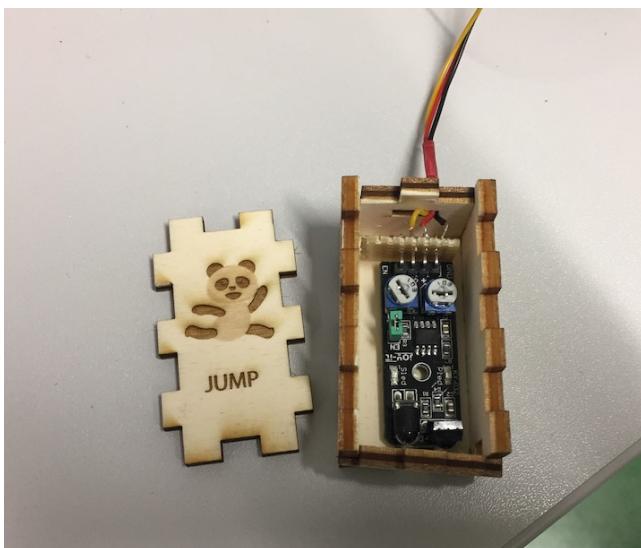
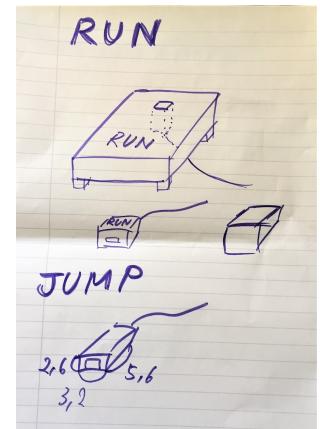
Start- und Ziellinie

Am Anfang und am Ende der Spindel haben wir auch zwei solcher Kontakte gebaut. Diese werden aber in beiden Höhenpositionen erreicht, also sowohl "während" eines Sprungs als auch "am Boden". Damit kann das Programm erkennen ob sich die Spielfigur am Ziel, oder auch am Start, befindet. Außerdem haben wir noch einen Schalter eingebaut, der die Spielfiguren wieder an den Startpunkt fährt, falls man während des Spiels abbrechen will. Für die Steuerung der Motoren und der Servomotoren haben wir vier Hindernissensoren mit unserem Arduino verbunden. Die, falls ein Hindernis erkannt wird, entweder den Motor der Gewindespindel bewegt oder den Servomotor der Spielfigur auf und ab fährt. Dabei haben wir über den Code gesteuert, dass der Sensor auch wieder unterbrochen werden muss, damit die Figur sich weiter bewegen kann. Damit wird ausgeschlossen, dass man nur etwas vor den Sensor hält und die Spielfigur "durchläuft".

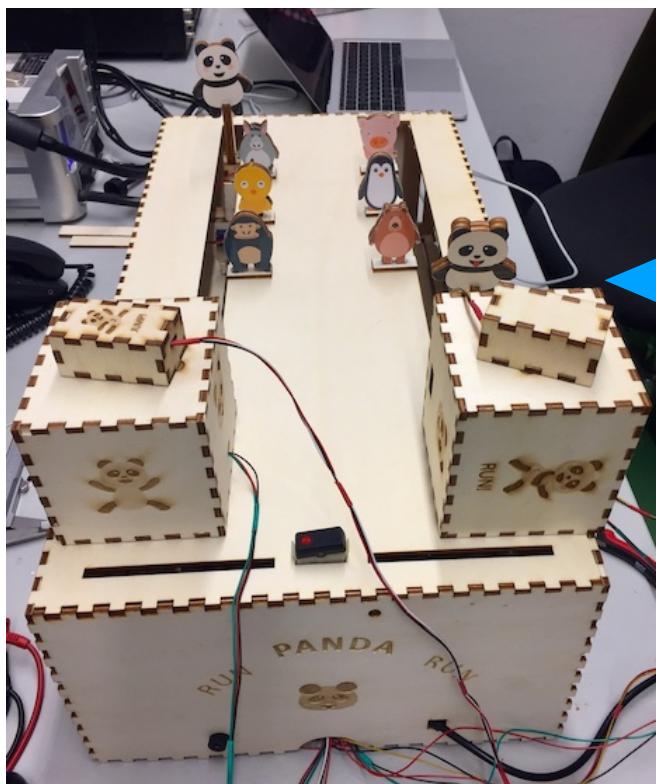
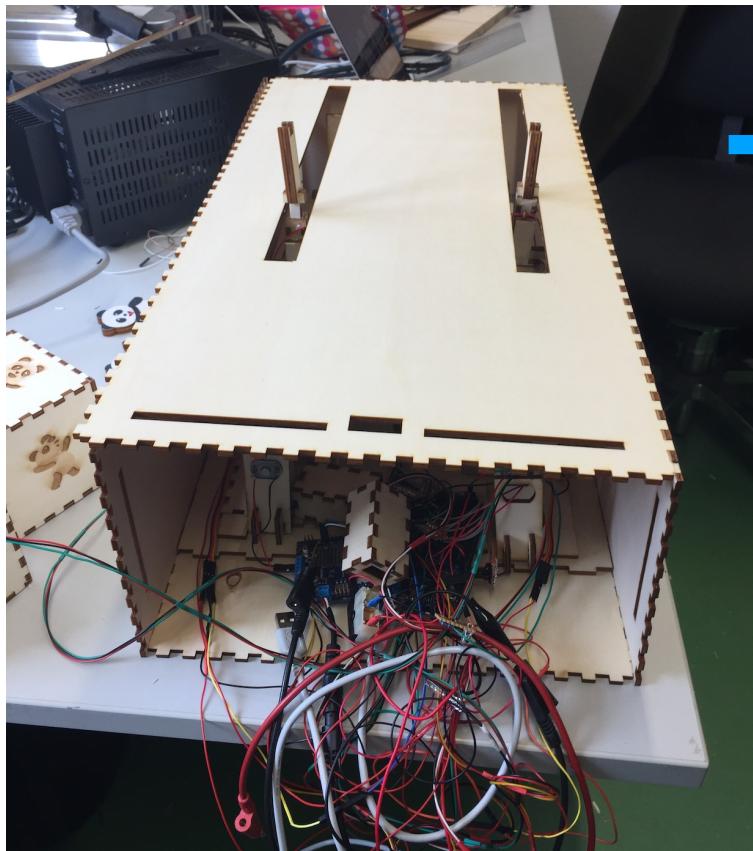


Controller mit Hinderniserkennung-Sensor

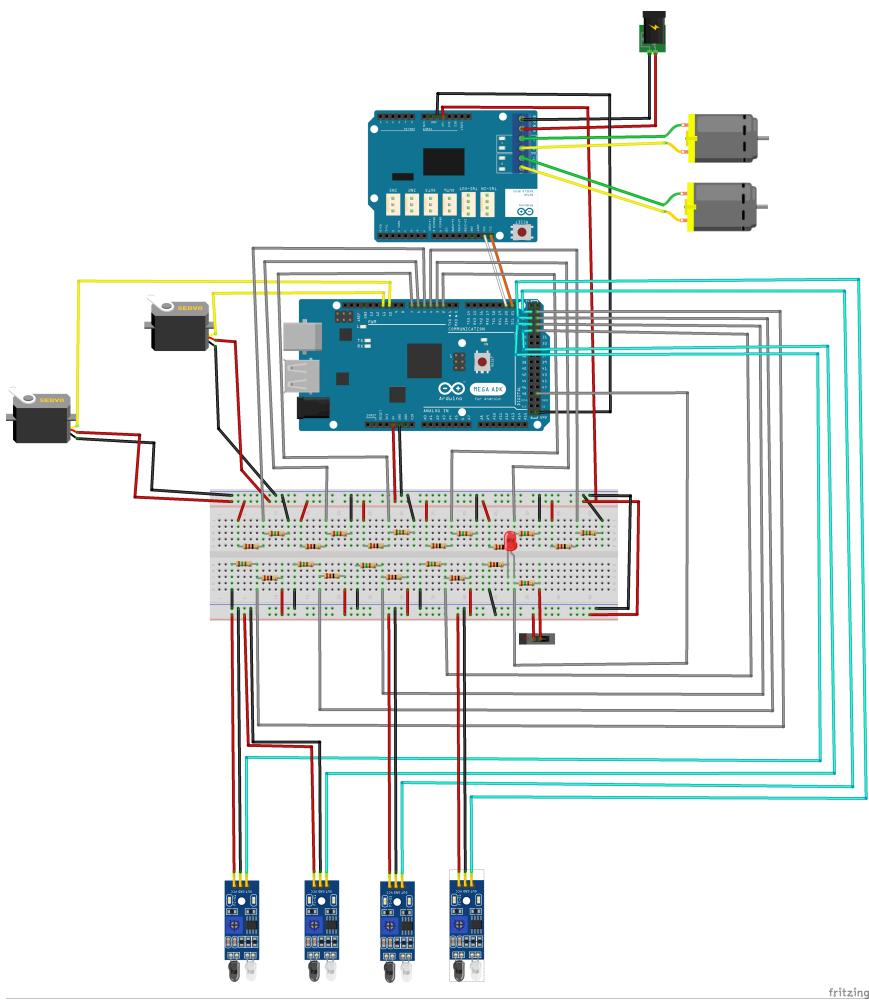
Hinderniserkennungs-Sensoren wurden für die Controller benutzt. Das Prinzip ist das: jedesmal, wenn sich ein Hindernis in der Nähe befindet, wird ein Signal geliefert. Unser Ziel ist, dass sich die Spieler möglichst schnell bewegen. Deswegen wird die Spielfigur jedesmal ein kleines Stück weiter vorgesetzt wenn der Sensor ein neues Hindernis erkennt. Dabei ist wichtig, dass der Sensor ein neues Hindernis erkennt, denn sonst würde ein permanent vor den Sensor gehaltenes Hindernis dazu führen, dass die Spielfigur bis zum Ziel durchläuft. Realisiert wurde dieses Verhalten mit einer einfachen Boolean-Variable "CanRun", die sich mit dem Sensorwert ändert.



Fertigstellung (Kisten, Hindernis- und Spielfiguren, etc.)



Verschaltungsdiagramm



Source Code

[Und hier noch der Source-Code für das Spiel.](#)

The Final Result

[Und hier noch das Video von 'RUN PANDA RUN' dazu.](#)