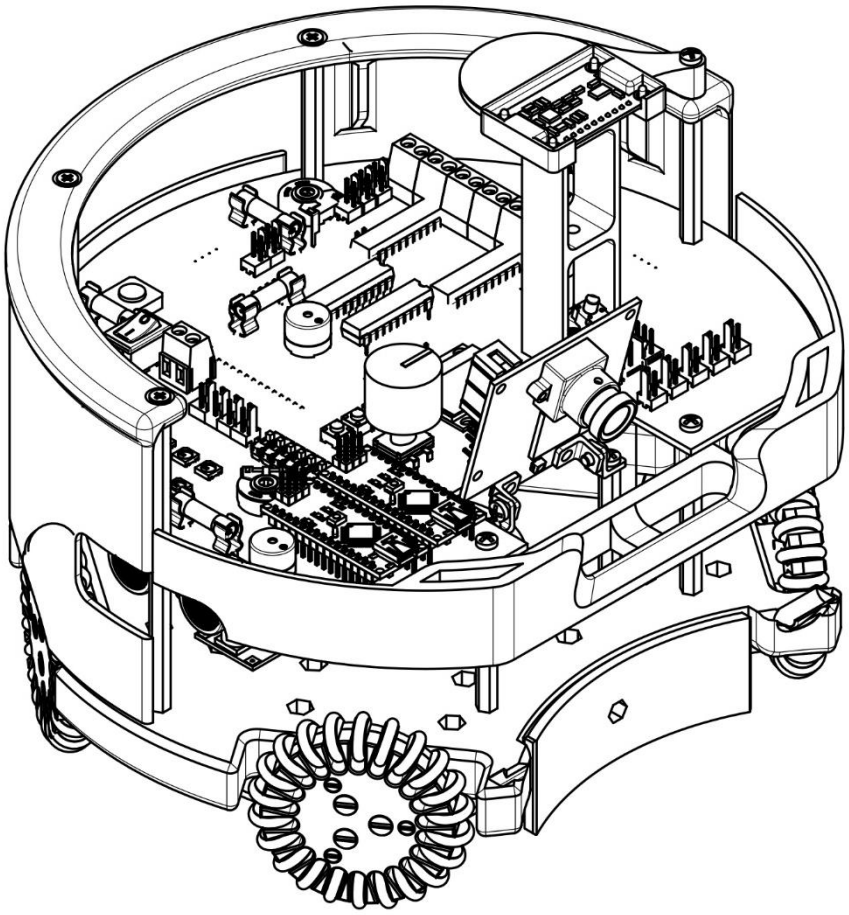
Dokumentation Iceberg Robots



Inhaltsverzeichnis

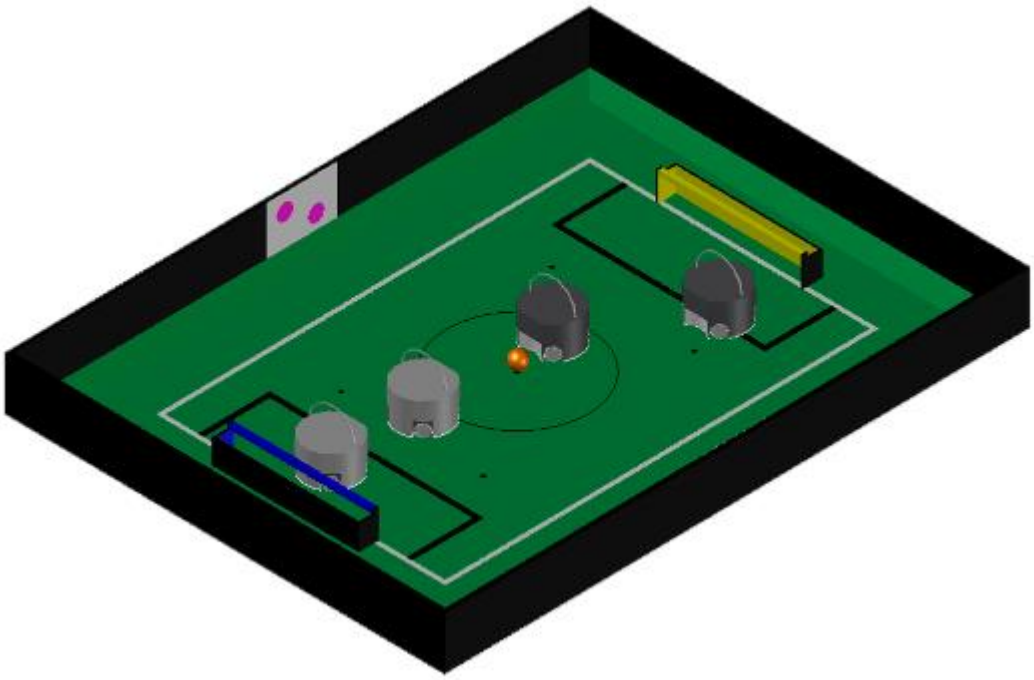
1. Wettbewerb

Der RoboCup Junior ist ein internationaler Wettbewerb für bis einschließlich 19-jährige, bei dem Teams Roboter designen, konstruieren und programmieren. Diese spielen in einer der drei Kategorien OnStage, Rescue oder Soccer gegen andere Teams.

Der Wettbewerb hat internationale Reichweite und es nehmen Schüler aus den verschiedensten Ländern am RoboCup Junior teil.

Die Iceberg Robots nehmen an der Kategorie Soccer Open League, teil. Hierbei müssen wir zwei Roboter konstruieren, welche autonom auf einem 1,82 m x 2,43 m großen Spielfeld Fußball spielen. Gespielt wird mit je 2 Robotern pro Team und einem farbigen Ball, welchen die Roboter nur mit Kameras erkennen können.

Das Spielfeld ähnelt einer verkleinerten Version eines menschlichen Fußballfeldes.



Spielfeldüberblick

1.1. Spielfeld

Abmessungen des Spielfelds

Eine weiße Linie trennt das Spielfeld in einen Innenbereich und einen Außenbereich. Die Roboter dürfen während des Spiels den 1,22m x 1,83m großen Innenbereich nicht vollständig verlassen. Das Ziel der Roboter ist es, Tore zu schießen, indem sie den orangefarbigen 0,074m großen Ball in das gegenüberliegende 0,6m breite Tor befördern. Auf dem Spielfeld befinden sich 5 schwarze Punkte. Diese werden neutrale Punkte genannt. Sie befinden sich in jeder Spielfeldhälfte rechts und links vor dem Strafraum und in der Mitte des Spielfelds. Um den Mittelpunkt liegt ein 0,6m großer Anstoßkreis. Das Spielfeld umrahmt eine 0,2m hohe schwarze Wand. In der Mitte der seitlichen Wände befindet sich eine Markierung aus zwei magentafarbenen Kreisen. Diese kann die Orientierung auf dem Feld erleichtern, denn sie ist mit der Kamera erkennbar. Vor den Toren markiert eine schwarze Linie den Strafraumbereich.

1.2. Spielfluss

Eine Partie besteht aus zwei 10-minütigen Halbzeiten. In der Halbzeitpause tauschen die beiden Teams die Seiten, um faire Bedingungen zuzusichern. Während Spielunterbrechungen wird die Spielzeit angehalten. Bei einem Anstoß werden alle Roboter gestoppt und von den Teams in der eigenen Spielhälfte im Innenbereich platziert. Hat ein Team ein Gegentor erhalten, so darf es einen Roboter in den Anstoßkreis setzen. Kein anderer Roboter dürfen diesen Bereich berühren.

Wenn sich der Ball 3 Sekunden nicht bewegt und es keinen Spielfortschritt gibt, ist der Spielfluss unterbrochen und die Schiedsrichter legen den Ball ohne Spielunterbrechung auf einen neutralen Punkt, sodass kein Team bevorzugt wird.

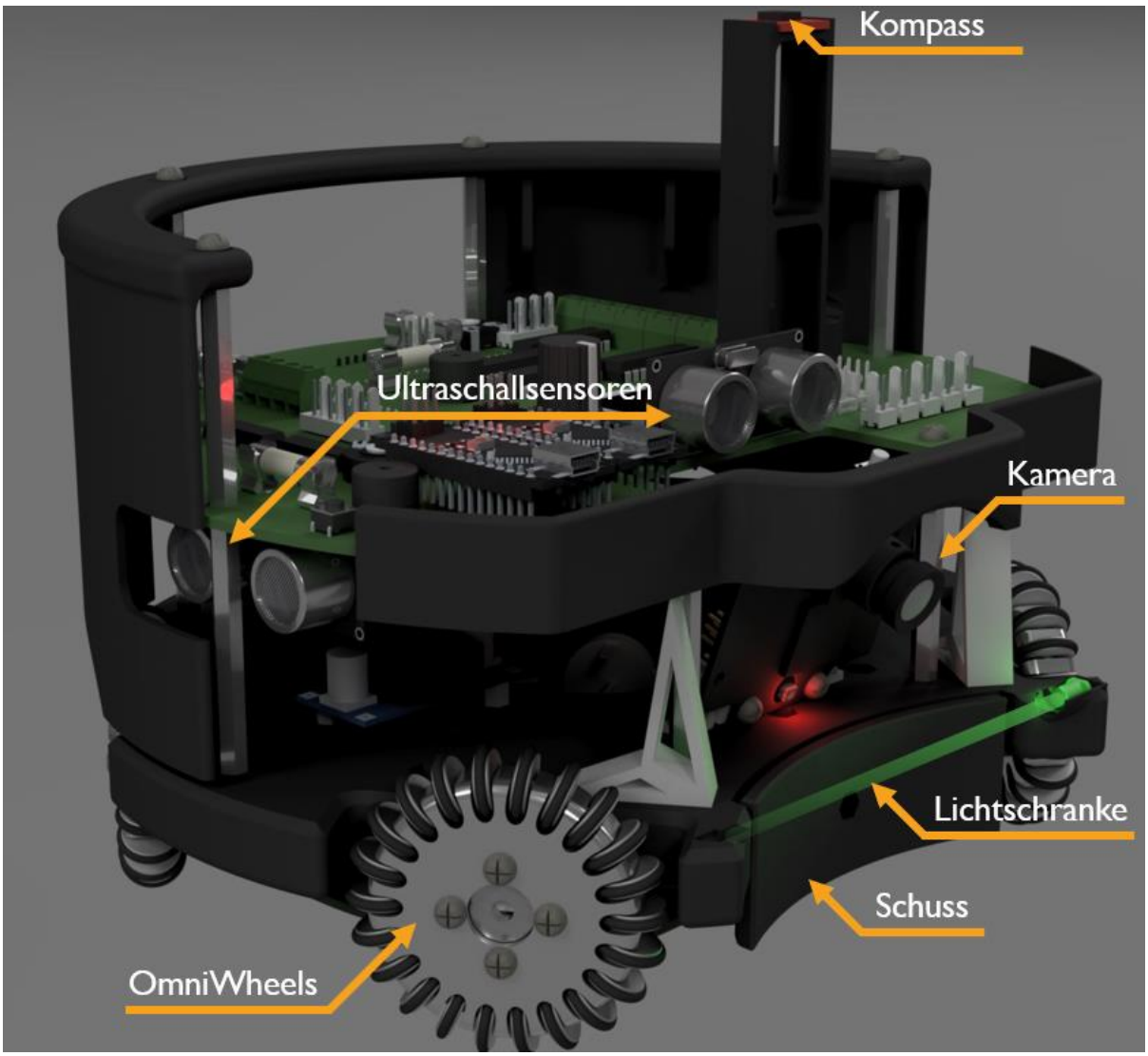
Es dürfen sich nicht beide Roboter des Teams in eigenen Strafraum befinden. Ist dies der Fall, setzen die Schiedsrichter den Roboter aus dem Strafraum heraus.

Verlässt der Roboter den Innenbereich ohne herausgeschoben zu werden, so muss er eine 1-minütige Strafzeit außerhalb des Spielfeldes verbringen.

Gelangt ein Ball gegen die innere Rückwand des Tores und wurde dabei der verteidigende Roboter im Strafraum nicht berührt, so gibt es ein Tor und einen Anstoß des Gegners.

2. Hardware des Roboters

Die Iceberg Robots haben seit April 2017 zwei neue baugleiche Roboter konstruiert. Erstmals planten wir den Roboter komplett in einem 3D-Modell bevor wir anfingen ihn zu bauen. Dieses Modell entstand in OnShape. Am Boden befindet sich ein dicker fast massier gedruckter Plastikkörper. Dieser schützt den Roboter beim Spiel vor Schäden bei Zusammenstößen. Außerdem befestigt der 3D Druck die Platine mit den Bodensensoren, fixiert die Motoren und sorgt dafür, dass alle Räder gleichen Bodenkontakt haben. Hätte ein Rad weniger Bodenkontakt, so hat der Roboter beim Fahren immer eine Eigenrotation, die die Steuerung stört. Darüber befindet ein Freiraum. Hier ist der Akku eingebaut. An der Vorderseite befindet sich die Kamera. Durch eine Klappe an der Rückwand des Roboter lässt sich der Akku in Sekundenschnelle wechseln. Darüber befindet sich die Hauptplatine. An ihr sind Ultraschallsensoren in alle 4 Richtungen befestigt. Eine Stützstruktur hält den Kompass an der höchsten Position des Roboters, denn die Magnetfeldsensoren sollten nicht von den Motoren gestört werden.



3D Modell des Roboters

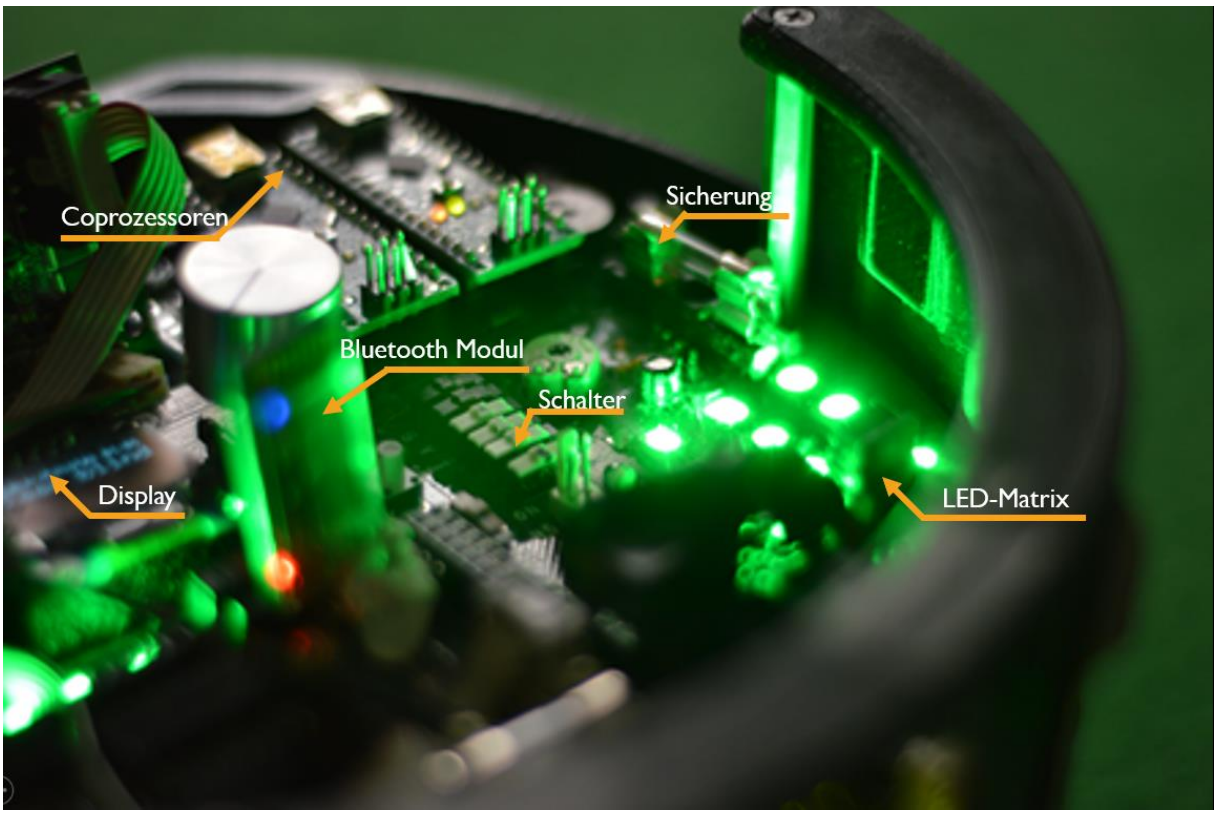
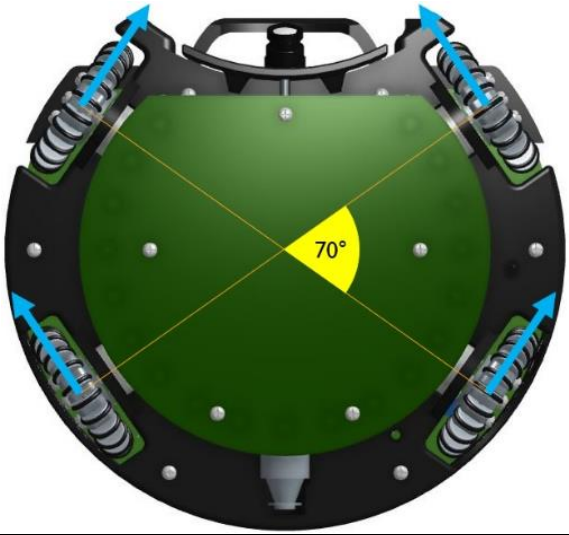


Bild der Roboter Oberseite

Auf einige wichtige Elemente gehe ich noch etwas konkreter ein:

4 Motoren: MAXON 12V 14W DC
Die beiden Motorachsen sind im 70° Winkel angeordnet, um beim Vorwärtsfahren schneller zu sein. Das ermöglicht uns auch bei einem gegnerischen Anstoß schnell am Ball zu sein. Die Kraft der Räder ist hier als Vektor dargestellt. Mit unserer selbstentwickelten Motorbibliothek können wir in jede beliebige Richtung fahren, jedoch mit verschiedener Maximalgeschwindigkeit. Dank des geringen Achsenwinkels blieb uns genügend Platz für den Solenoid, den wir als Schuss nutzen.



4 selbst designte Omni-Wheel Räder
Omni-Wheel Räder bestehen aus einem großen Rad, welches über die Motorachse angetrieben wird. Diese Achse ist hier eingezeichnet. Außerdem hat es viele kleine frei bewegliche Unterräder. Die Unterräder sind auf einem Metalldraht befestigt.

16 RGB Leds mit Helligkeitssensoren

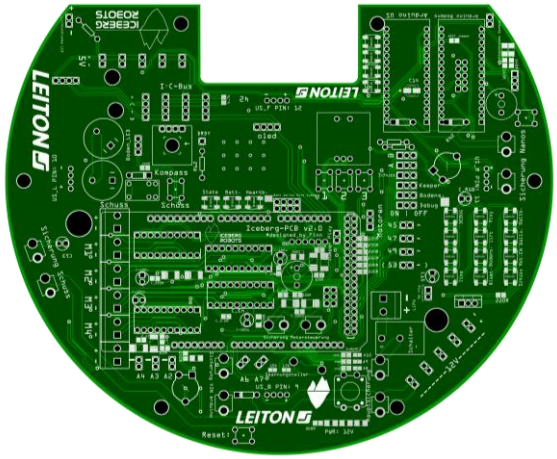
Am Boden des Roboters befindet sich eine Platine mit 16 ringförmig angeordneten RGB Leds. Diese sind einzeln ansteuerbar und haben jeweils einen Helligkeitssensor. So können wir bestimmen, ob und wo wir eine Linie berühren. Eigentlich ist rotes Licht für uns optimal, den es reflektiert auf dem grünen Flies kaum und stellt so einen starken Kontrast zu den weißen Linien dar. Leider kann dieses rote Licht von anderen Teams mit dem Ball verwechselt werden. Um diese Situation zu lösen, können wir einfach über einen Schalter auf der Hauptplatine zwischen weißem und rotes Licht auswählen. Und zum Spaß haben wir auch noch eine sich drehende Farbanimation, wie in der Abbildung zu sehen, programmiert.

---

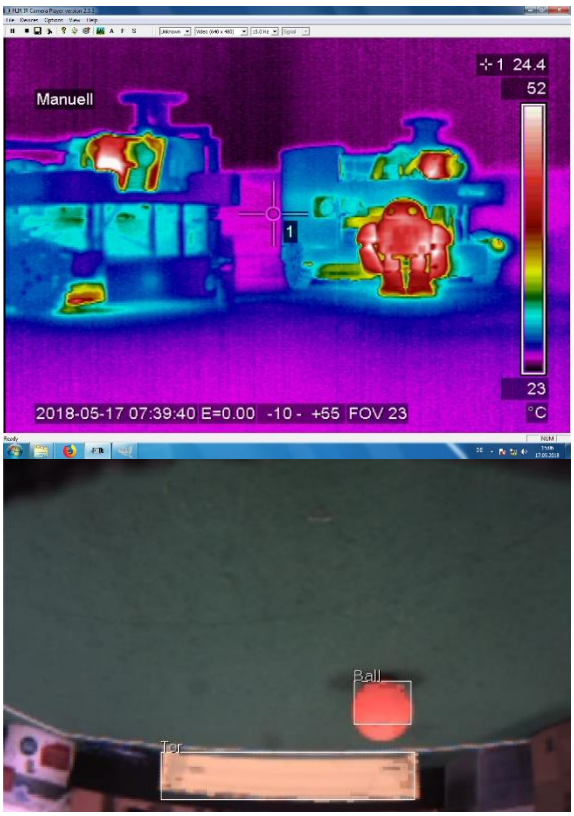
1 Hauptplatine

Auf der Hauptplatine befinden sich:

- alle 3 Arduinos
- Sicherungen
- 15 RGB Leds, um Zustände von Sensoren oder des Programms anzuzeigen
- OLED-Display mit Scrollrad, um durch verschiedene Tabs zu scrollen
- Schalter
- Motortreiber
- etc.



1 CMUcam5 Pixy Kamera
Wir nutzen diese Kamera, weil sie einen Prozessor für die Bildverarbeitung direkt integriert hat. Um sie auf den Ball und das Tor einzustellen, schießen wir sie an den Computer an und stellen die Farbbereiche ein. Auf dem Infrarotbild ist gut zu erkennen, wie heiß unsere Kamera wird. Die nötige Rechenleistung der Bildauswertung ist mit Arduinos nicht zu erreichen, weshalb die Kamera auch deutlich mehr Strom als die Arduinos verbraucht und heißer wird. Das untere Bild zeigt die Auswertung des Kamerabildes. Die Kamera sendet Informationen über die Boxen, in denen sich der Ball bzw. das Tor befinden, an den Arduino Mega.



3. Strategie

3.1. Kamera

Eine Box enthält Informationen über ihre Position (x,y-Koordinaten), ihre Größe(Höhe und Breite) und ihre Objektnummer(Ball, Tor oder Colorcodes).

Wenn wir mindestens einen Ball sehen, betrachten wir nur die Box mit dem größten Flächeninhalt. Analoges gilt für die Tore und die Colorcodes. Colorcodes sind farbige Markierungen an der Rückwand der Roboter, damit sich die Roboter gegenseitig erkennen können.

3.2. Rollenwechsel

Die Roboter sind via Bluetooth verbunden und legen so dynamisch die beiden Rollen Verteidiger und Stürmer fest. So können wir auch sicherstellen, dass nur ein Roboter in den eigenen Strafraum fährt.

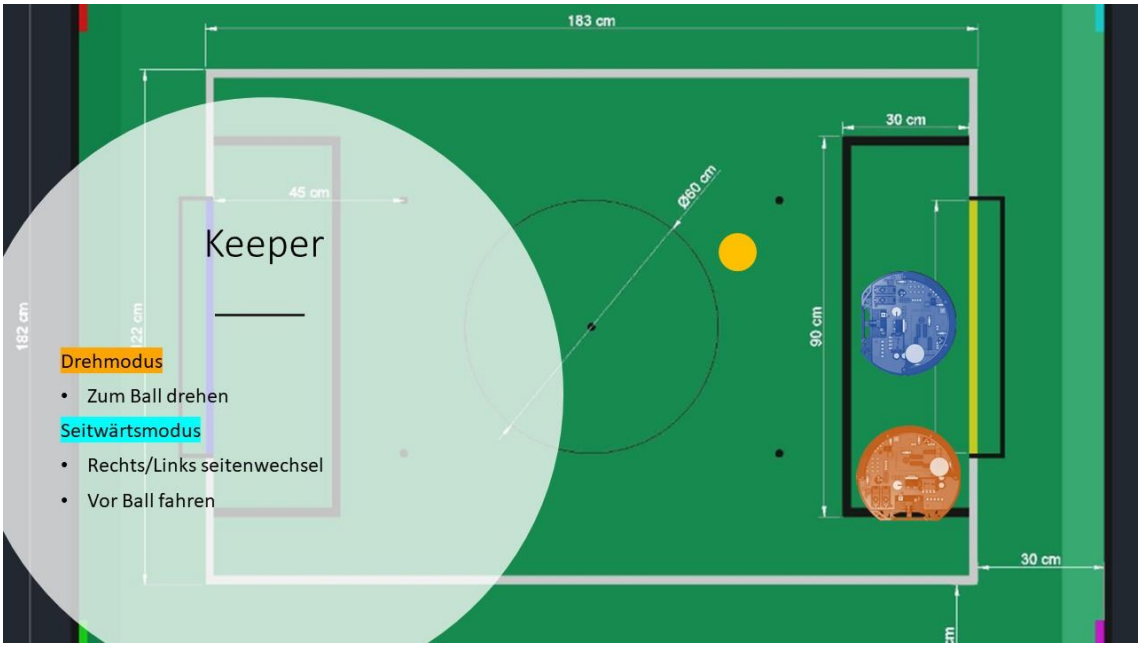
Beide Roboter berechnen mit einer Bewertungsfunktion eine Punktzahl. Wenn sich die Punktzahl um einen bestimmten Schwellwert unterscheidet, so werden die Rollen synchron verändert. Dies ist auch visuell zu erkennen, denn der neue Verteidiger(Keeper) blinkt grün und der Stürmer(Rusher) magentafarben auf.

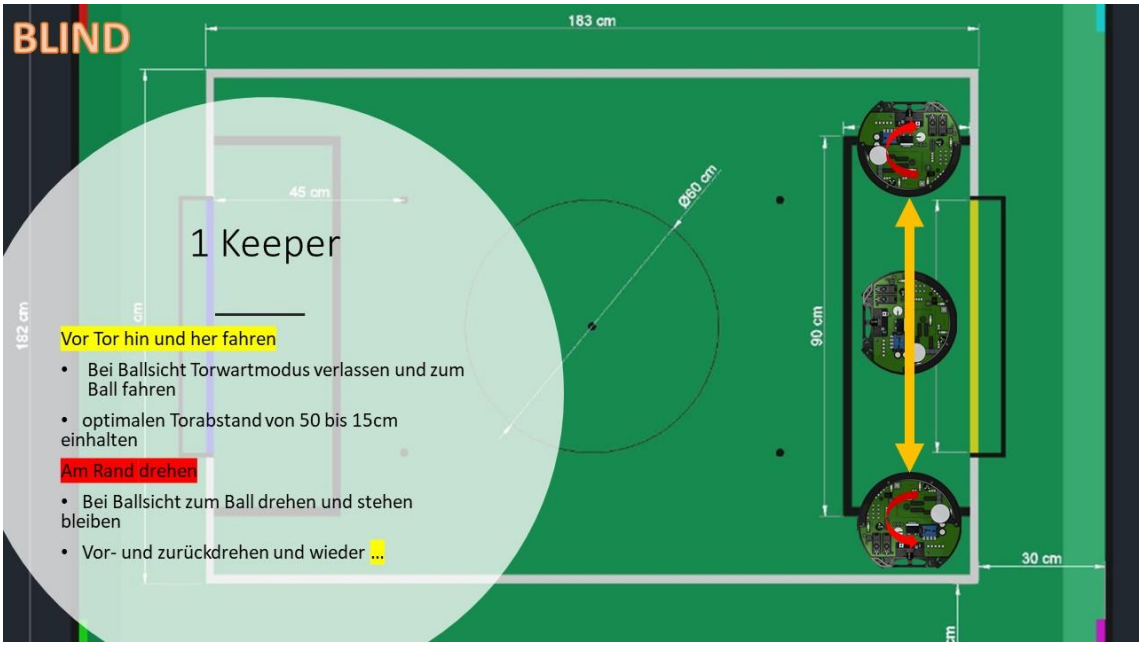
Die Bewertungsfunktion soll umso höher sein, desto besser die Position des Roboters für eine Offensive ist. Deshalb werden verschiedene Aspekte unterschiedlich gewertet.

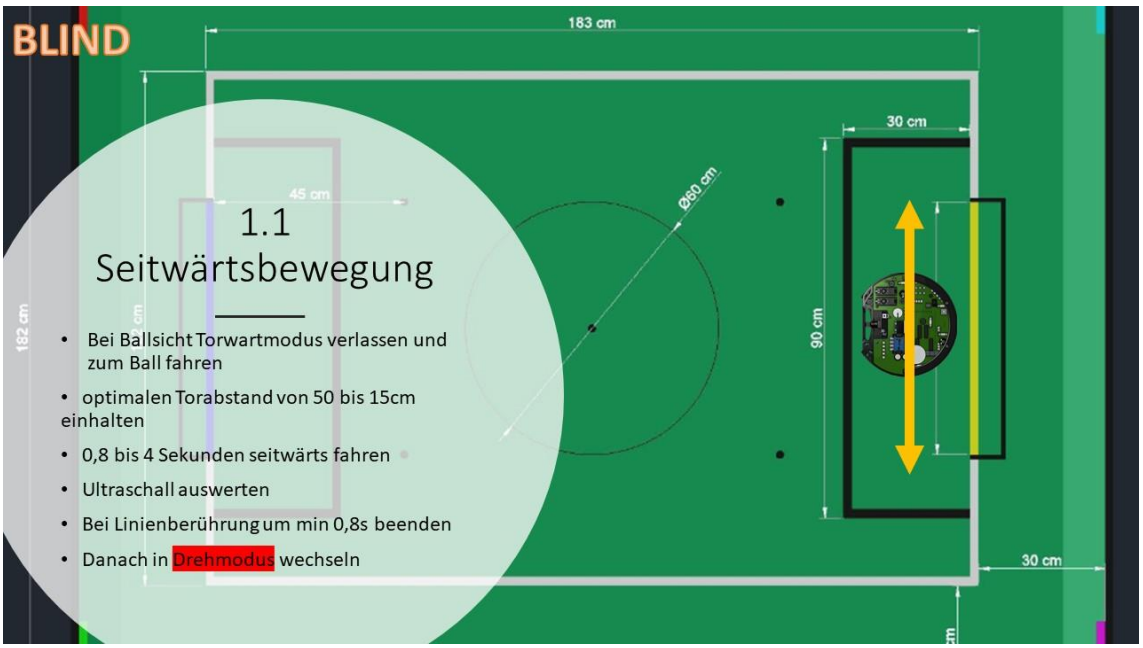
Die Boxbreite des Balls ist umso größer, je näher der Ball ist. Wenn wir nah am Ball sind können wir schneller Tore schießen. Deshalb wird sie mit der Stärke 100 gewichtet. Außerdem ist es günstig, wenn sich der Ball gerade vor dem Roboter befindet. Der Ballwinkel wird mit der Stärke 50 gewichtet. Des Weiteren ist es günstig, wenn wir der vordere Roboter sind, denn dann sind wir näher am Tor. Der Abstand nach hinten wird mit der Stärke 30 gewichtet. Sollten wir das Tor sehen, können wir einen gezielten Richtungsschuss machen. Die Torsicht wird mit der Stärke 10 gewichtet.

3.3. Blindzustand

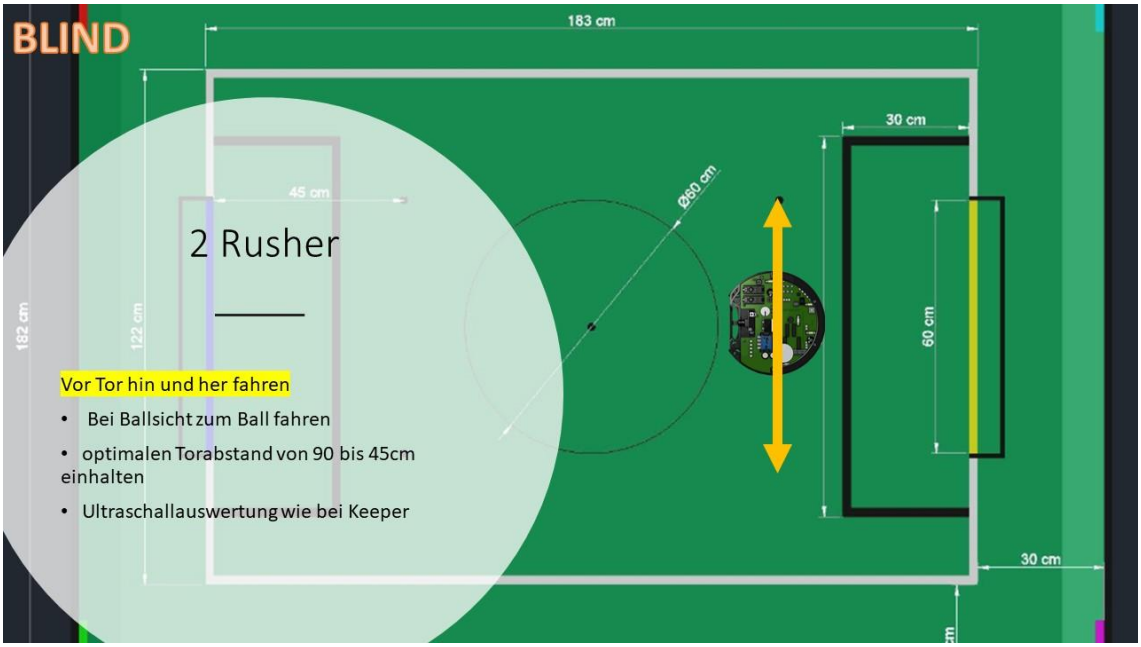
Wenn der Roboter keinen Ball sieht, verteidigt er das Tor, indem er seitlich hin und her fährt und sich an den Torpfosten dreht, um sich umzusehen.

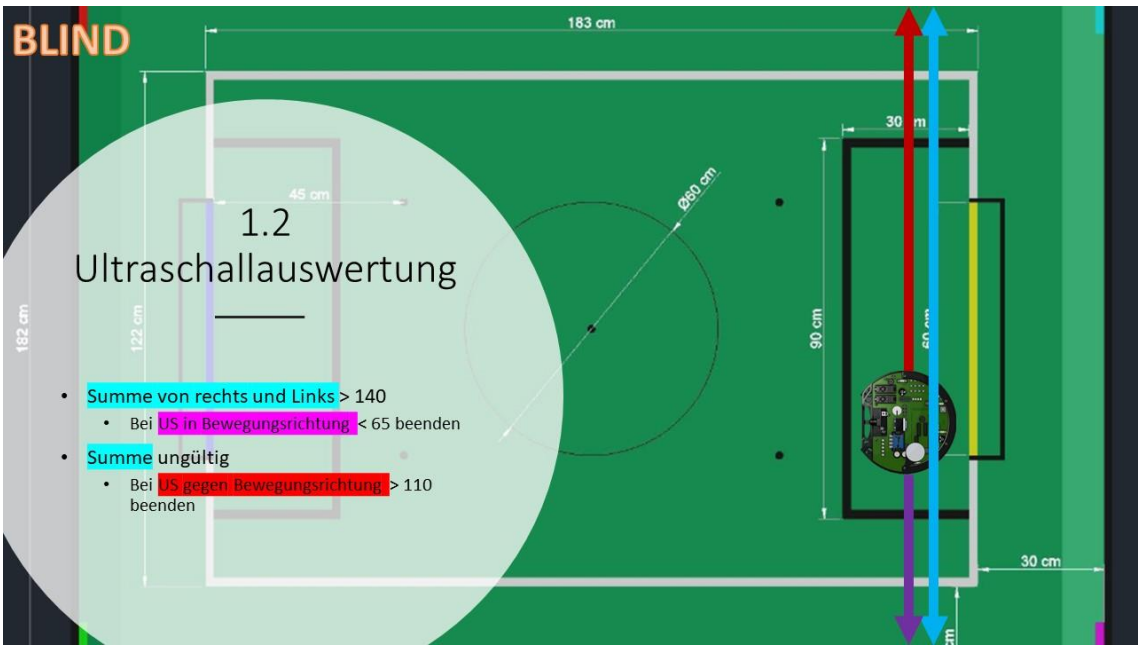




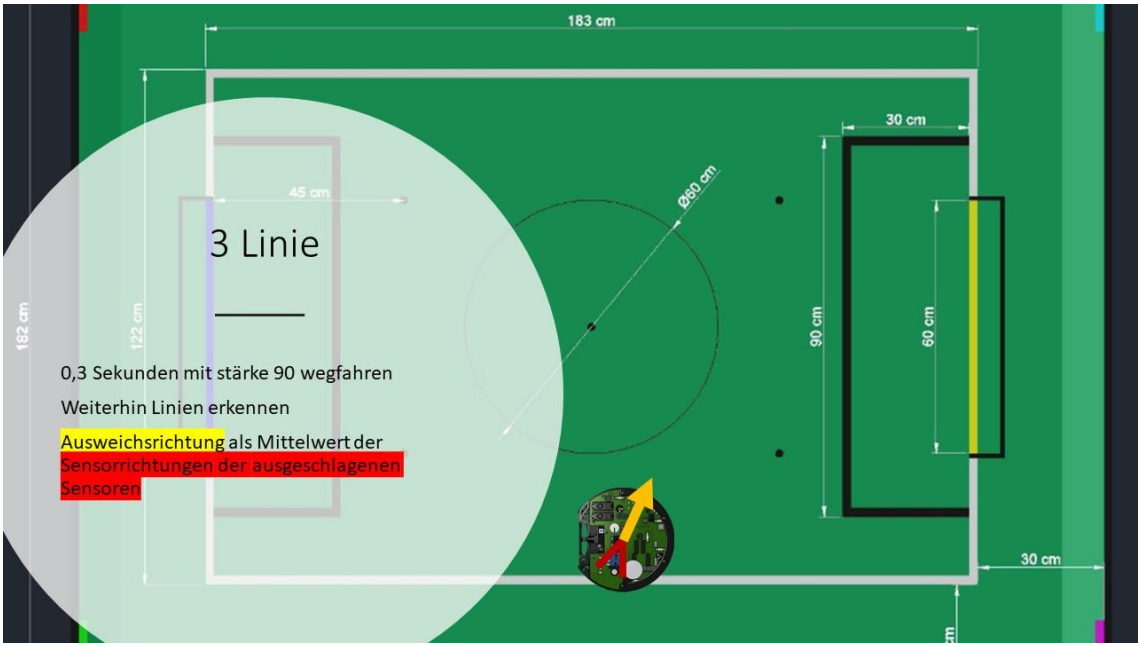


Doch nur der Verteidiger darf in den Strafraum fahren. Sollte der Stürmer den Ball nicht sehen, fährt er vor dem Strafraum seitlich hin und her ohne sich zu drehen.

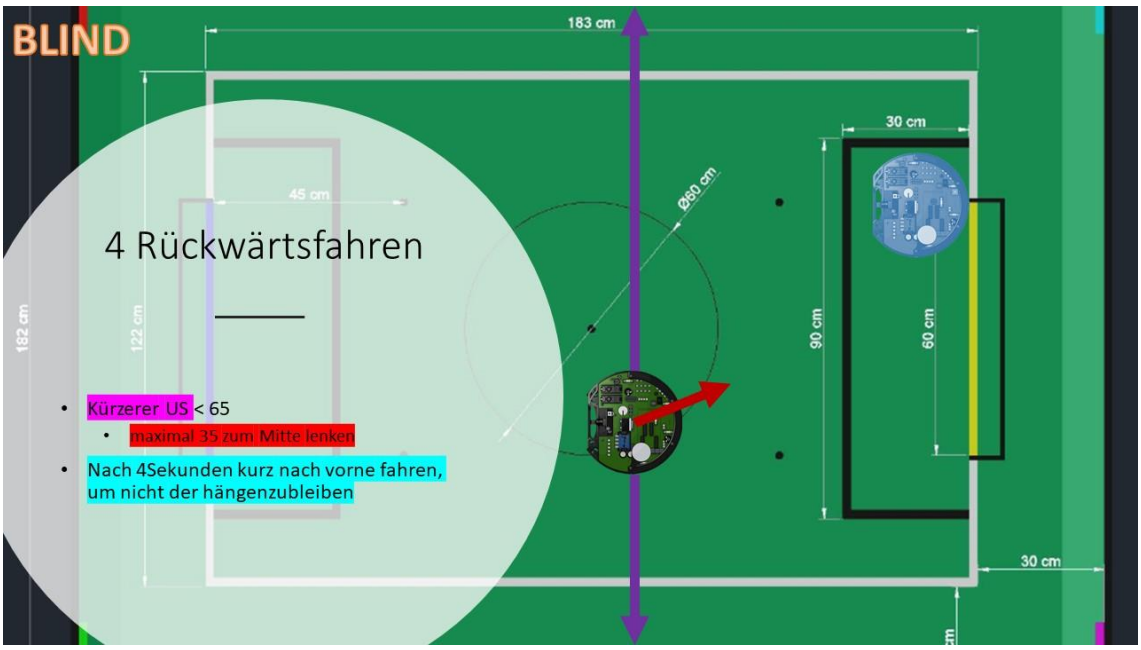




Die Linie hat eine höhere Priorität als der aktuelle Strategiezustand. Erkennt der Roboter eine Linie, so berechnet er die Position der Linie und fährt in die entgegengesetzte Richtung, um noch im Spielfeld zu bleiben.

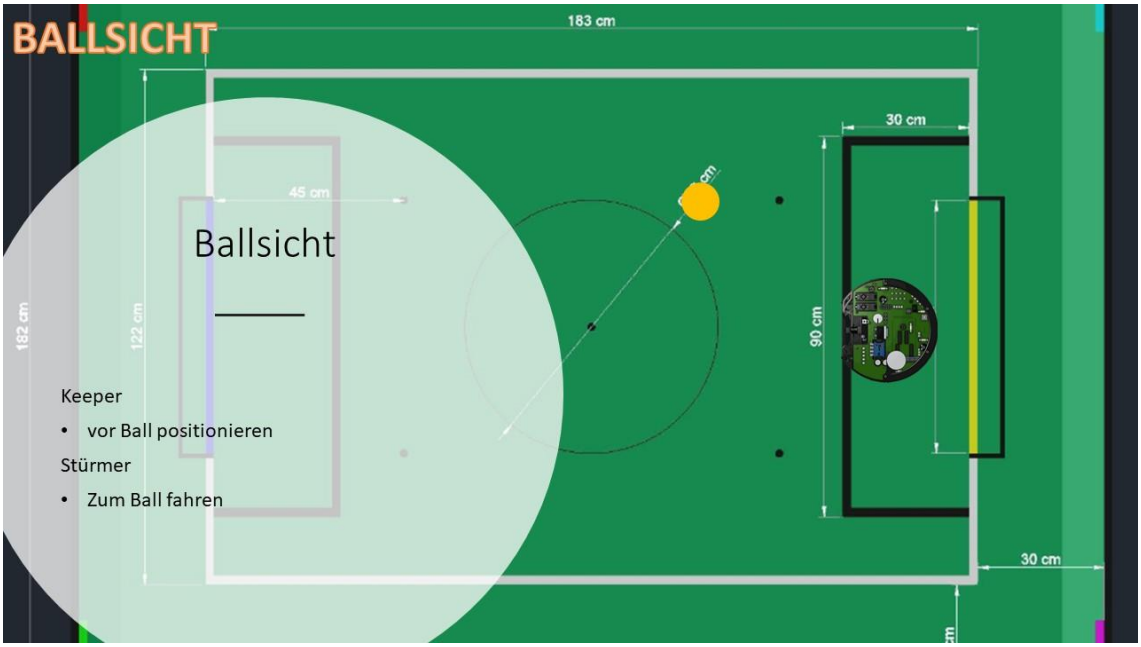


Sollten wir im Blindzustand zu weit nach vorne gelangen, so fahren wir wieder zurück vor das eigene Tor. Wir nutzen dabei das Tor als Hindernis, um zu bremsen.

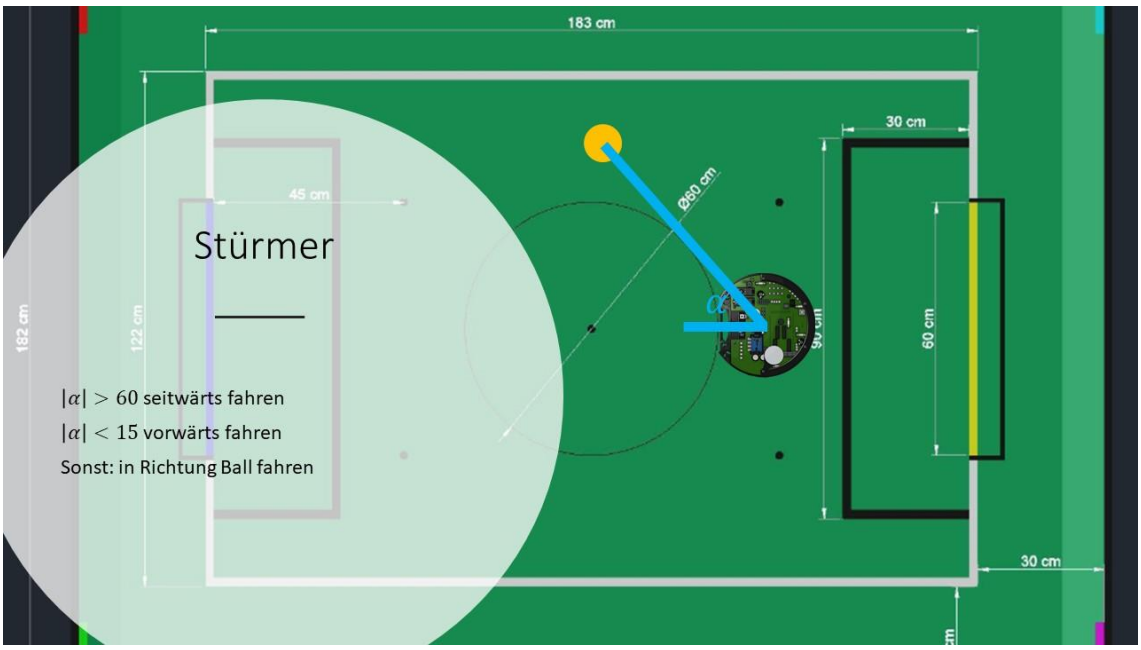


3.4. Ballsicht

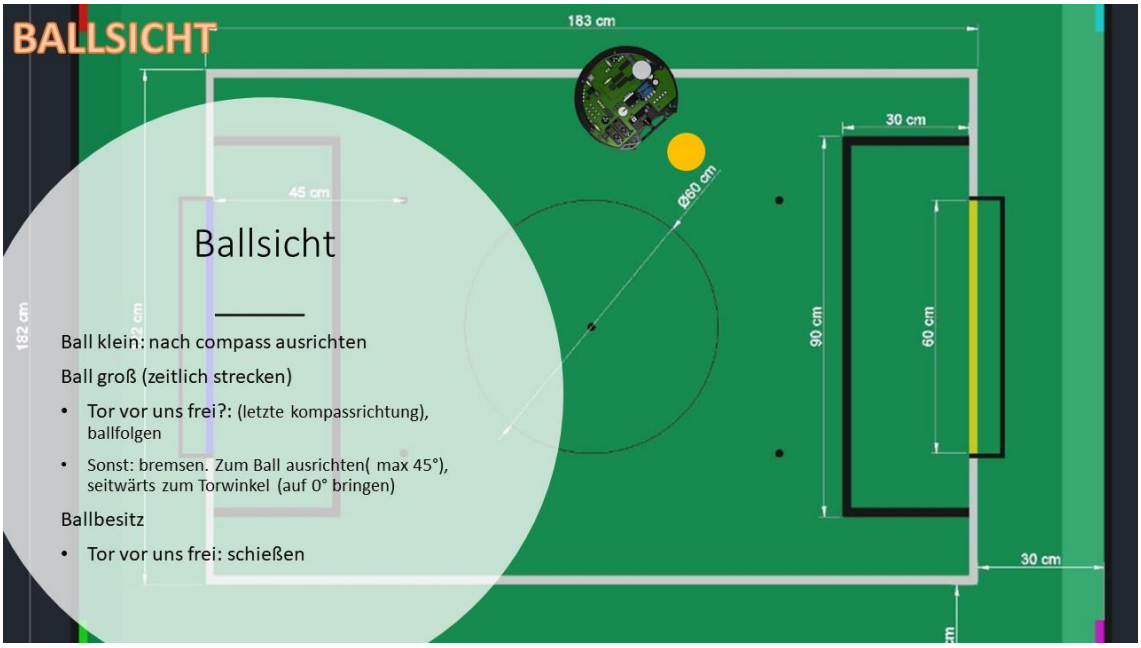
Der Stürmer beginnt eine Offensive, wenn er den Ball sieht. Der Verteidiger soll jedoch weiter das Tor verteidigen. Deshalb bleibt er vor dem Tor, aber positioniert sich zwischen dem Ball und dem Tor.



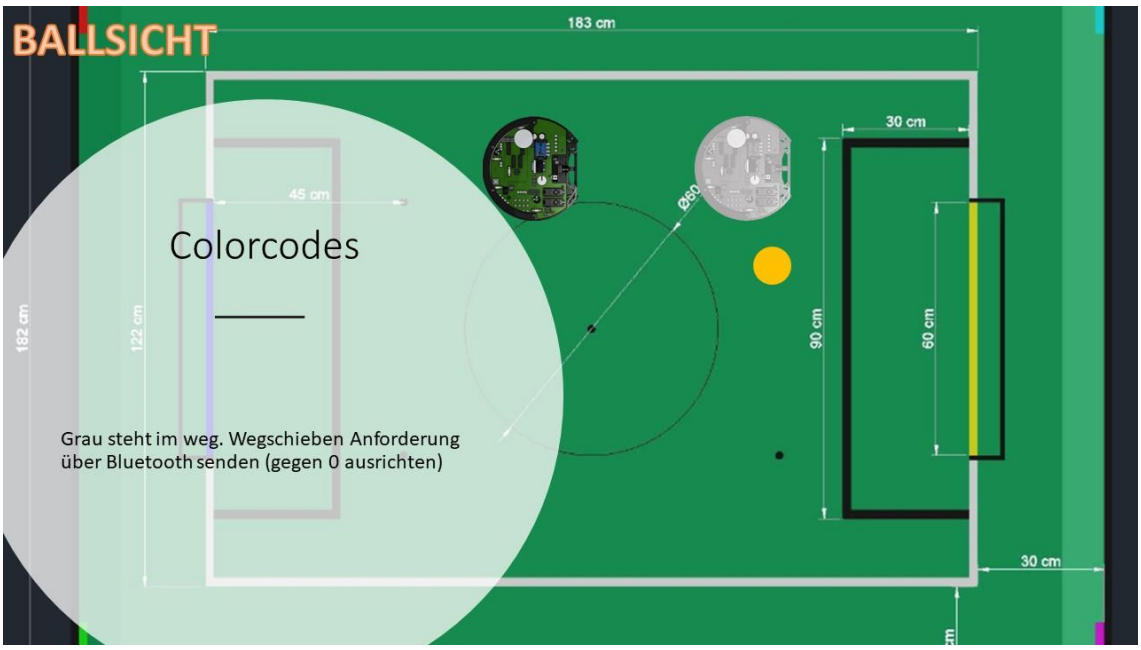
Unser Roboter hat aufgrund der Achsenanordnung Probleme eine 70° Fahrrichtung auszuführen ohne dabei seine Orientierung zu verändern. Unser Kompass sorgt dafür, dass sich unserer Orientierung beibehält, aber die dafür notwendigen Korrekturbewegungen verlangsamen den Roboter. Deshalb fährt er bei einem großen Ballwinkel seitwärts im 90° Winkel.



Wenn wir den Ball und das Tor sehen, versuche wir uns, den Ball und das Tor auf eine Linie zu bringen. So können wir auch vom Spielfeldrand das Tor treffen. Ist der Ball nah genug, bremsen wir kurz und drehen uns dann nach rechts oder links um den Ball, um die optimale Position zu erlangen.



Damit die Roboter vor allem bei einem Rollenwechsel nicht gegeneinanderstoßen, sendet der Stürmer den Verteidiger einen Befehl nach rechts bzw. nach links auszuweichen. Die Roboter sehen sich gegenseitig mithilfe der Colorcodes.



4. Software des Roboters

Das Hauptprogramm des Arduino Mega besteht aus 18 Tabs:

- **iceberg.ino**

  Hier wird die setup und loop Methode definiert. Alle Globalen Variablen werden hier initialisiert. Die Kommunikation mit den Coprozessoren bzw. dem anderen Roboter wird initialisiert. Es werden alle anderen Programmabschnitte geregelt:

  Die Leds werden eingestellt. Über Funk werden Motoren gestartet oder gestoppt. Zwischen Verteidiger und Stürmer wird gewechselt (je nachdem wer mehr bewertungspunkte hat). Alle 30ms wird die Kamera abgefragt. Alle 1000ms wird der Bildschirm aktualisiert.

- **Config.ccp, Config.h**

  Hier werden alle Einstellungen definiert. Wie lange soll ein bestimmter Zustand durchgeführt werden? Welche Daten sollen über die serielle Kommunikation mit dem Computer übertragen werden?

- **Display.cpp, Display.h**

  Das OLED-Display wird angestuert. Es gibt folgende Tabs mit Unterpunkten:

Home	Sensor	Bewertung	Kamera
☐ Spielrolle(Stürmer/Verteidiger)	☐ Ultraschallwerte	☐ Punkte insgesamt	☐ Grafische Darstellung der erkannten Boxen
☐ Fahrrichtung	☐ Lichtschranke	☐ Punkte Ballbreite	
☐ Statusinfo	☐ Akkuspannung	☐ Punkte Ballmittigkeit	
	☐ Linienrichtung	☐ Punkte Abstand nach hinten	
	☐ Beschleunigungssensor	☐ Punkte Torsicht	
	☐ ...		

- **Led.cpp, Led.h**

  Ansteuerung der Led Matrizen und der Lautsprecher.

- **Mate.cpp, Mate.h**

  Bluetooth Kommunikation

- **Pilot.cpp, Pilot.h**

  Motorsteuerungsbibliothek

- **Player.cpp, Player.h**

  Intelligente Zustandsmaschine

- **Ultrasonic.cpp, Ultrasonic.h**

  Abstandsmessung

- **Utility.cpp, Utility.h**

  weitere Methoden wie Kamera auslesen.

- **pin.h**

  Alle Pin Anschlüsse