

Einleitung

Ziel dieses Projekts ist es, mit NAO Robotern die Spiele Tic Tac Toe und Vier Gewinnt an einem Tablet zu spielen. Dabei soll sowohl Roboter gegen Roboter, als auch Roboter gegen Mensch spielen können. Die Aufgabenbereiche: Spielstanderkennung, Spieltaktik, Bewegung des Roboters sowie eine zugehörige App mit Multiplayer.

Spieltaktik

Die Spieltaktik erhält von der Vision das aktuelle Spielfeld, welches auf dem Tablet angezeigt wird. Anhand des Felds kann berechnet werden, auf welches Feld der Roboter nun klicken sollte.

1 Tic Tac Toe:

X	O	
	X	
		O

Aus diesem Feld wird dann der bestmögliche Zug berechnet. Dabei können für Tic Tac Toe verschiedene Regeln aufgestellt werden, um zu überprüfen, welches das bestmögliche Feld ist. Beispielsweise wird überprüft, ob zwei eigene Zeichen nebeneinander sind, weil dann das 3. Feld dazu gesetzt werden kann, um zu gewinnen.

2 Vier Gewinnt:

		Y				
	R	Y	R			
	Y	R	Y	Y	R	
R	Y	Y	R	R	Y	R

Bei Vier Gewinnt wird für jede Spalte berechnet, wo ein weiterer Spielstein landen würde. Danach wird für jede Richtung (horizontal, vertikal und diagonal) für die beide Seiten (z.B. horizontal-links, horizontal-rechts) und die dortigen 3 nächsten Felder berechnet, wie viele eigene Spielsteine und wie viele freie Felder nebeneinander liegen, ohne durch einen Stein vom Gegenspieler vom aktuellen Feld getrennt zu sein. Damit weiß man wie wichtig dieses Feld ist, um offensiv zu spielen. Gleichzeitig berechnet man dasselbe für die Defensive. Aus dieser Berechnung ergibt sich eine Priorität für offensive und defensive Spielzüge für jede der sieben Spalten. Zusätzlich wird für jedes Feld berechnet, wie wichtig das darüber liegende Feld für den Gegenspieler ist.

Am Ende gibt die Spieltaktik den nächsten Zug zurück.



Bewegungen

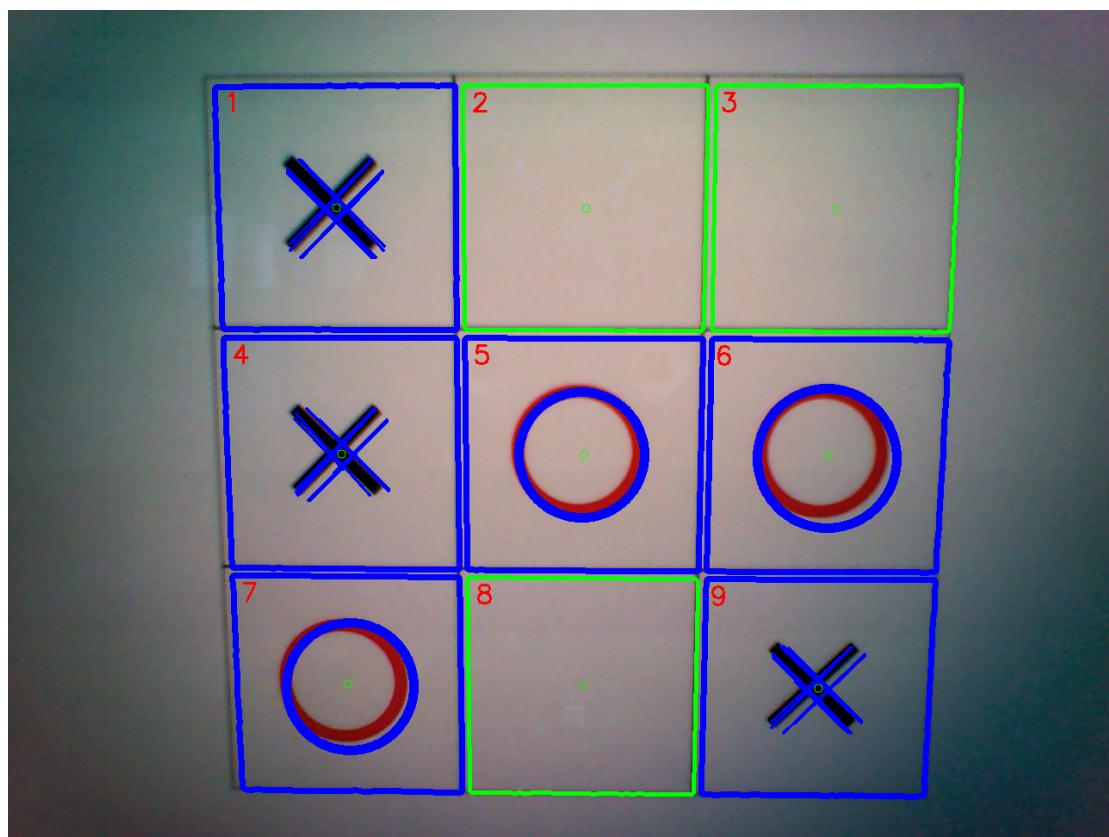
Damit Klicks auf dem Tablet ausgelöst werden können, müssen die Plastikfinger, Energie leiten können. Dafür erhalten die NAOs Aluminium Fingerkuppen. Die Bewegungen des Roboters funktionieren über eine Methode des Motion Proxys, die die naoqi bereitstellt. Dieser kann übergeben werden, welcher Motor in welche Position eingestellt werden soll. Durch eine Abfolge an Bewegungen kann der Roboter somit Klicks ausführen. Konkret sind für Klicks aufs Tablet 81 Positionen für eine bestimmte Positionierung des Tablets manuell eingestellt. Diese kann man sich wie ein über den Bildschirm gelegtes Gitter mit 9 Linien horizontal und 9 vertikal vorstellen. An jedem Schnittpunkt ist eine dieser manuell erhaltenen Positionen. Wenn also auf die Position genau in der Mitte des Tablets geklickt werden soll, wird die Positionen mit für Höhe 5 und für Breite 5 verwendet. Für Klicks zwischen diesen beispielsweise Höhe 4,5 und Breite 5 wird interpoliert, also aus den Werten für Höhe 5 mit Breite 5 und Höhe 4 mit Breite 5 wird die Höhe 4,5 berechnet. Es wird quasi um den gewollten Punkt auf dem Tablet das naheliegendste Rechteck mithilfe der 81 Punkte gezogen. Aus den Bewegungen der vier Punkte, die dafür benötigt werden, wird die neue Bewegung für den neuen Punkte berechnet.

Damit diese manuellen Klicks funktionieren, muss das Tablet immer genau gleich stehen. Durch bestimmte Positionierungen der Arme des Roboters wird dem Benutzer gezeigt, wie das Tablet vor dem Roboter hingestellt werden muss. Dabei müssen Höhe, Abstand und Kippungen des Tablets korrekt sein.

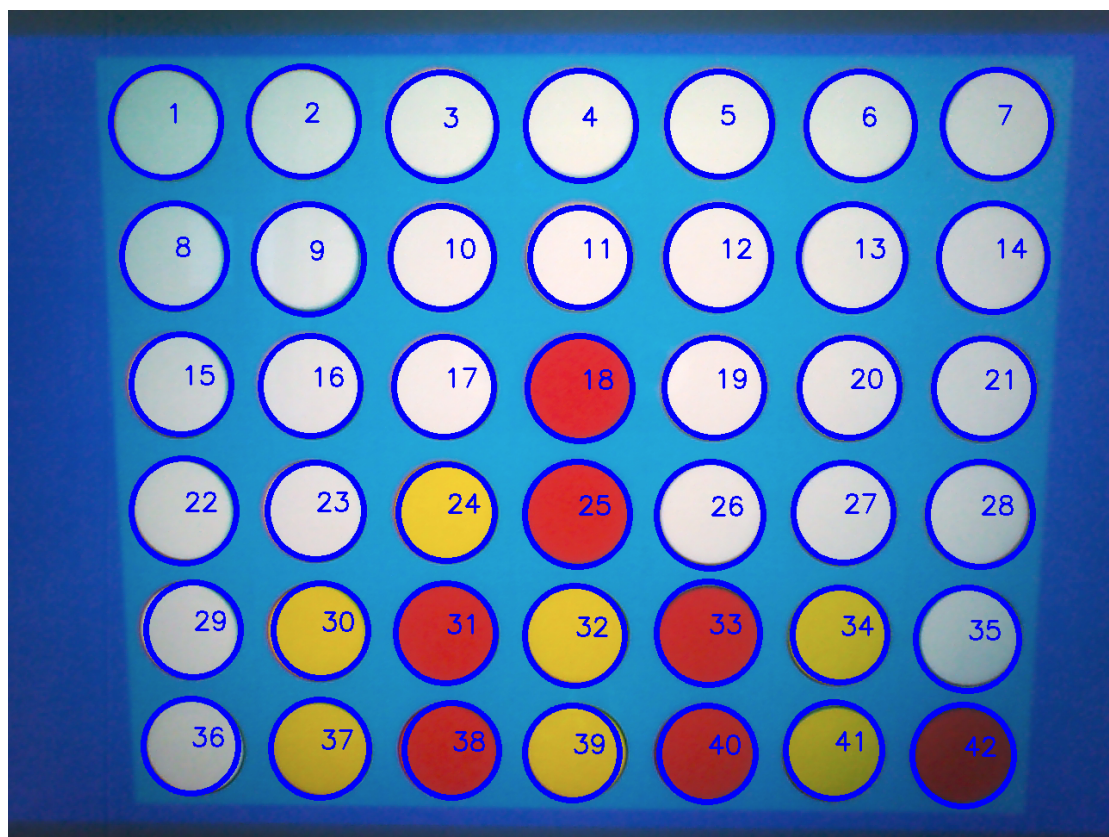
Spielstanderkennung

Die Erkennung des Spielfelds und des Spielstandes erfolgt über etablierte *Computer Vision* Algorithmen. Zu Beginn wird das aufgenommene Bild mit Gaussfilter und *Canny*-Kantendetektion vorverarbeitet.

Tic Tac Toe Die *Tic Tac Toe*-Erkennung nutzt den Aufbau des Spielfelds, indem es über die hierarchische Segmentierung nach [1] das Feld in die neun Teilfelder unterteilt. Werden exakt neun erkannt, so wird mittels Hough-Transformation bestimmt, ob jeweils ein Kreis oder Kreuz vorliegt[2].



Vier Gewinnt Die *Vier Gewinnt*-Erkennung nutzt ebenfalls den Aufbau des Spielfelds. Werden mithilfe der Hough-Transformation $7 \times 6 = 42$ Kreise erkannt, muss im Anschluss nur noch die Farbe innerhalb jedes Kreises bestimmt werden.



4. App

Die App wurde mithilfe von Flutter entwickelt und verfügt sowohl über einen Singleplayer als auch einen Multiplayer. Unsere Entscheidung für Flutter basiert auf dessen plattformübergreifender Natur und der großen Auswahl an UI-Komponenten. Im Multiplayer haben Benutzer die Möglichkeit, innerhalb desselben lokalen Netzwerks miteinander Tic Tac Toe oder Vier Gewinnt zu spielen.

Der Multiplayer basiert auf dem Host-Client Prinzip, wo eines der Geräte die komplette Kontrolle über die Spielauswahl und das Fortfahren nach Spielende bestimmt, während das andere Geräte den Anweisungen des Hosts folgt. Der Verbindungsaufbau läuft über ein simples Handshake-Verfahren, um eine erfolgreiche Verbindung zu garantieren. Danach wird auf Seite des Hosts mit **RouteAwareWidgets** und auf Seite des Clients mit dem **ClientRoutingService** das automatische Navigieren durch die App ermöglicht und ein neues Spiel kann begonnen werden.

Die Anwendung wurde speziell an die Kamera und das Bewegungsmuster des Roboters angepasst und ist auf Android, IOS, Linux und Windows lauffähig.

Literatur

[1] Satoshi Suzuki and Keiichi Abe. "Topological structural analysis of digitized binary images by border following." In: *Computer Vision, Graphics, and Image Processing* 30.1 (1985), pp. 32–46. URL: <http://dblp.uni-trier.de/db/journals/cvgip/cvgip30.html#SuzukiA85>.

[2] Reinhard Klette. *Concise Computer Vision - An Introduction into Theory and Algorithms*. Undergraduate Topics in Computer Science. Springer, 2014, pp. 1–413. ISBN: 978-1-4471-6319-0. URL: <http://dx.doi.org/10.1007/978-1-4471-6320-6>.