Hochschule für angewandte Wissenschaften Kempten

**Rocky Hockey IV**

*BILD MIT NEUEM GESTELL*

*Abschlussbericht*

*Entwicklung eines Air Hockey Roboters*

[Fakultät](http://lukas-mertens.de/) Informatik

[Sommersemester](http://lukas-mertens.de/) 2020  
  
  
  
  
07.07.2020

|  |
| --- |
| Zusammenfassung |
| Das Projekt Rocky Hockey ist ein DV-Projekt, das unter der Leitung von Professor Arnulf Deinzer für Studierende im 6. Semester des Studiengangs Informatik angeboten wird. Ziel des Projekts ist es einen Roboter zu Entwickeln, der gegen einen Menschen in einem “Airhockey Match“ antreten kann. |

|  |  |  |
| --- | --- | --- |
| Projekt-Team | | |
| *Name* | *Kontakt* | *Matrikelnummer* |
| *Roman Wecker* | roman.w.wecker@stud.hs-kempten.de | 360659 |
| *Alexander Fichtl* | alexander.m.fichtl@stud.hs-kempten.de | 364931 |
| Thomas Brücklmayr | thomas.bruecklmayr@stud.hs-kempten.de | 369884 |
| Marco Schwärzler | marco.k.schwaerzler@stud.hs-kempten.de | 326906 |
| André Fischer | andre.p.fischer@stud.hs-kempten.de | 366231 |
| Thomas Gantner | thomas.gantner@stud.hs-Kempten.de | 366212 |
| Meiko Mehnert | meiko.mehnert@stud.hs-kempten.de | 360589 |
| Sarah Hörmann | sarah.m.hoermann@stud.hs-kempten.de | 366941 |
| David Lippert | david.c.lippert@stud.hs-kempten.de | 363233 |
| Dominik Veith | dominik.veith@stud.hs-kempten.de | 374134 |

|  |  |
| --- | --- |
| Projekt-Betreuung | |
| Prof. Dr. Arnulf Deinzer | arnulf.deinzer@hs-kempten.de |

Inhaltsverzeichnis

[Zusammenfassung 1](#_Toc44682737)

[1. Projekt Ist-Zustand 3](#_Toc44682738)

[1.1 IST – Zustand Software 3](#_Toc44682739)

[1.2 IST – Zustand Hardware 4](#_Toc44682740)

[2. Codeänderungen 6](#_Toc44682741)

[2.1 Dokumentation und Softwarearchitektur der JAVA GUI 6](#_Toc44682742)

[2.2 Neue Implementierungen 6](#_Toc44682743)

[2.3 Integration der ArduCam Module 6](#_Toc44682744)

[3. Hardwareänderungen 7](#_Toc44682745)

[3.1 Hebefunktion der Tischplatte 7](#_Toc44682746)

[3.2 Neues Kameragestell 7](#_Toc44682747)

[3.3 ArduCam Module 7](#_Toc44682748)

[3.4 Elektrik und Verkabelung 7](#_Toc44682749)

[3.5 Aufbau der Zwischenplatte 7](#_Toc44682750)

[4. Probleme 8](#_Toc44682751)

[5. Ausblick Team V 9](#_Toc44682752)

[6. Anleitungen und Tipps 10](#_Toc44682753)

[Literaturverzeichnis 11](#_Toc44682754)

# 1. Projekt Ist-Zustand

1.1 Einleitung

Es ist ein Nachbau eines Air-Hockey Tisches, wie man ihn aus Spielhallen kennt, gegeben. Dieser Nachbau besteht aus einem Gehäuse, in der sich die Elektronik des Tischs versteckt, eine Spielplatte zum Herausheben und einem Gestell für eine Festmontierte Kamera sowie einem Anzeigebildschirm für den Punktestand.

Eine Seite des Tisches kann von einem menschlichem Spieler bespielt werden, wohingegen die andere Seite durch einen festmontierten Roboter gesteuert wird. Der Roboter verwendet die über dem Tisch montierte Kamera. Die Kamera überblickt die Spielhälfte des Roboters und erkennt den Puck auf der Spielplatte. Die Kamera gibt das Bild weiter an die Software die aus dem gewonnen Bild neben der Flugbahn auch die Geschwindigkeit des Pucks sowie mögliche Verteidigungspositionen berechnet.

## IST – Zustand Software

Die Software des Projekts umfasst C# Module sowie eine JAVA – GUI

Unser Repository:

<https://github.com/R-Studios/Rocky-Hockey-IV>

🡪 Sollten glaube dem Prof auch einen Zugang geben damit das würde die Übergabe zum nächsten Projektteam deutlich beschleunigen und wir hätten nicht so viel Stress im nächsten Jahr

1.1.1 C# - Abschnitte

GoalDetectionFramework:   
Erfasst die Tore, sobald die Pucks die Lichtschranken überschritten haben.

MotionCaptureFramework übermittelt einen Stream der Kamera. Aus den gewonnen Bilder wird über das MoveCalculationFramework die Flugbahn sowie die Geschwindigkeit des Pucks berechnet und gleichzeitig eine Verteidigungsstrategie berechnet. Im MovementFramework werden die Motoren angesteuert die jeweiligen Befehle für die Bewegungen stammen hierbei aus dem MoveCalculationFramework.

1.1.2 JAVA – GUI

Die verwendete GUI, die über den Monitor am Gestell verwendet wird0

1.1.3 Repository

Der Projektcode wurde uns in mehreren Branches über eine Dropbox übergeben. Ein bestehendes Repository war nicht vorhanden

## IST – Zustand Hardware

**Raspberry Pi:**

Der Raspberry Pi übernimmt die Darstellung der GUI und ermittelt die Tore über die Laserschranken.

**Monitor:**

Über den Monitor wird eine GUI abgerufen über die man ein Spiel starten kann. Während eines laufenden Spiels zeigt der Monitor außerdem den Punktestand an.

**Kamera:**

Bei der Kamera handelt es sich um eine Playstation 3 Eye Kamera die via Motion-Tracking das Spiel aufzeichnet. Die Kamera befindet sich montiert an einem Gerüst auf der Spielseite des Roboters.

**Laserschranken:**

Laserschranken in beiden Toren zeichnen auf ob ein Tor geschossen worden ist.

**Motoren:**

Die Bewegungen des Roboters werden über zwei Nema Stepper Motoren ermöglicht.

**Lüfter:**

Für das typische Gleiten beim Air-Hockey sorgt ein 90W ebm-papst Industrie Lüfter. Zu finden ist er auf der unteren Seite des Tisches.

Lautsprecher:

Über ein Soundboard werden Spielsounds (z.B. bei einem Tor) abgespielt.

# 2. Codeänderungen

## 2.1 Dokumentation und Softwarearchitektur der JAVA GUI

\*\*\* Roman Wecker \*\*\*

## 2.2 Neue Implementierungen

Zu Beginn des Projektes wurden einige Zeilen Code überarbeitet, um die Performance des MoveCalculationFramework zu verbessern, außerdem musste an vielen Stellen die Dokumenation des Codes verbessert werden, da hier nur ungenügende Kommentare vorhanden waren und Methoden und Vorgehensweisen nicht deutlich genug gekennzeichnet worden sind. Zudem wurden einige Dateien und Methoden die nicht dienlich waren aus dem Projekt entfernt.

Da der bisherige Code nur darauf aus war die Flugbahn des Pucks zu berechnen und auf die berechnete Flugbahn lediglich mit einem Block zu antworten haben wir uns entschlossen, dass MoveCalculationFramework dahingehend zu verändern, dass der Roboter auch aggressivere Manöver ausführen kann. Dazu wurde zuerst eine neue Idee für die Strategieberechnung umgesetzt. Hierbei wurde das Interface StrategyCommunicationInterface gebildet. Daraus entstanden StrategyManager und SimpleStrategy, ersteres wurde dazu verwendet die Berechnung für eine Strategie anzustoßen während zweiteres unterschiedlich viele Strategie berechnet. Festgelegt zu diesem Zeitpunkt war es, dass vier Strategien ermittelt werden sollen. Aus diesen Strategien wurde dann über die StrategyMaster diejenige gewählt die einerseits schnell genug war, um den Puck abzufangen und andererseits über eine X-Koordinate ausgewählt worden ist. Über die Methode execute() in StrategyManager kann nun die Strategie ausgeführt werden.

Mit der PathPrediction wird die alte Berechnung für die Flugbahn ersetzt. Vorteil Hierbei ist das auch Kantentreffer besser berechnet werden können und damit eine zuverlässigere Aussage über die Flugbahn getroffen werden kann. In der DirectionForStrategy wird eine Flugbahn für die Strategie berechnet, die den Puck auch möglicherweise über Kantentreffer zum Ziel schlägt.

TESTFRAMeWORK VON DOMI

Repository:

Als Repository haben wir in GitHub eine Projektgruppe erstellt für zehn Personen. Dazu wurde das alte Projekt auf einem Branch festgehalten während wir abseits vom Masterbranch auf refactored C# die Änderungen am alten Projekt bearbeiteten. Das Finale Projekt wurde dann über C#-Alpha-Version weiterentwickelt ehe es zum Schluss auf den Master gepusht wurde.

## 2.3 Integration der ArduCam Module

\*\*\* Alexander Fichtl \*\*\*

# 3. Hardwareänderungen

## 3.1 Hebefunktion der Tischplatte

\*\*\* Andre Fischer – Hebefunktion der Tischplatte \*\*\*

## 3.2 Neues Kameragestell

\*\*\* Sarah Hörmann \*\*\*

## 3.3 ArduCam Module

Seit diesem Semester steht außerdem eine Arducam CMOS AR0135 zur Verfügung, die die Playstatio3 Eye Kamera ersetzen soll.

\*\*\* Alexander Fichtl \*\*\*

## 3.4 Elektrik und Verkabelung

\*\*\* Meiko Mehnert \*\*\*

## 3.5 Aufbau der Zwischenplatte

\*\*\* Thomas Gantner \*\*\*

4. Probleme

\*\*\* ALLE \*\*\*

**Beispielabschnitt**

**Vorheriges Team**

Durch Herrn Prof. Dr. Deinzer war es möglich direkt zu Beginn des Semesters mit dem vorherigen Team in Verbindung zu treten. Jedoch war dabei der Austausch innerhalb der Teams sehr mühselig, da vor allem im vorhergehenden Team gerade einmal zwei ehemalige Studenten dem jetzigen Team zugearbeitet haben und Fragen beantworten konnten. Wir wurden teilweise mit offenen Fragen zurückgelassen und der Info wir sollen uns offene Fragen aus dem Abschlussbericht ausarbeiten. Was jedoch aufgrund der ungenügenden Dokumentation kein Spaß war. Ein Großteil der Anfangsphase war deswegen damit beansprucht, dass vor allem das Programmierteam sich in den Code einlesen musste und Zusammenhänge und Rückschlüsse selbst erarbeiten musste. Erschwerend dazu kam die unzureichende Dokumentation innerhalb des Codes hinzu.

5. Ausblick Team V

\*\*\* ALLE \*\*\*

**Beispielabschnitt**

Beispieltext

Hier sollte ausgeschrieben werden, welche Ziele wir verfolgt haben aber nicht beenden konnten. Bzw. welche Ansätze vom nächsten Team verfolgt werden sollten.

Reihenfolge möglichst nach Überschriften in Abschnitt 3 und 4.

6. Anleitungen und Tipps

\*\*\* ALLE \*\*\*

**Beispielabschnitt**

Beispieltext

V

Vollständige Angabe zur Ausarbeitung der ArduCam

Reihenfolge möglichst nach Überschriften in Abschnitt 3 und 4.

**Ausarbeitungen**  
Dokumenationen, Wochenberichte, die Abschlussberichte der Vorgängergruppen, das alte C# - Projekt, die JAVA GUI findet man alles in unserem Repository u.a. auch in den verschiedenen Branches. Zum Ende des Projekts sollte auch Herrn Deinzer ein Zugang dazu ermöglicht werden, damit eine Übergabe deutlich einfacher und schneller abgewickelt werden kann.

**Elektrotechniker**

Veränderungen an der Elektronik benötigen gewisse Erfahrungen. Sollte diese in eurem Team nicht vorhanden versucht doch über den Professor jemanden aus der Fakultät Elektrotechnik für euch zu gewinnen.

# Literaturverzeichnis

**Zitierweise**

Alle Zitate und Belege wurden nach dem APA-Zitierstandard verfasst. Wenn sich längere Abschnitte auf eine Quelle beziehen, wird immer nur einmal im Abschnitt auf die Quelle verwiesen.

Agarap, A. F. M. (2019, 7. Februar). *Deep Learning using Rectified Linear Units (ReLU)*. University of Cornwell, [arXiv:1803.08375](https://arxiv.org/abs/1803.08375) [cs.NE]. Abgerufen 2. Juni, 2019, aus der arXiv Datenbank.