

DARTS DETECTION

Automatisches kamerabasiertes Darts Scoreboard

Mika Paul Salewski

Hochschule für Angewandte Wissenschaften Hamburg

04.02.2025

AGENDA

- Aufgabenstellung
- Aufbau & Grundidee
- Komponenten und Struktur der Software
- Ergebnisse
- Schwierigkeiten
- Software

AUFGABENSTELLUNG

- Automatische Erkennung eines neuen Darts in der Dartscheibe
- Erkennung, dass 3 Darts geworfen und wieder entfernt wurden
- **Erkennung der genauen Position und Wert des Darts** (z. B. T19)
- Anzeigen der aktuellen Punktestands der Spieler („GUI“)
- Umsetzung einer Kommandozeile zur Konfiguration oder Korrektur des Spiels

AUFBAU & GRUNDIDEE

- Hardware-Aufbau:
 - LED-Ring mit 3 Kameras

TOP



RIGHT



LED-Ring

LEFT



AUFBAU & GRUNDIDEE



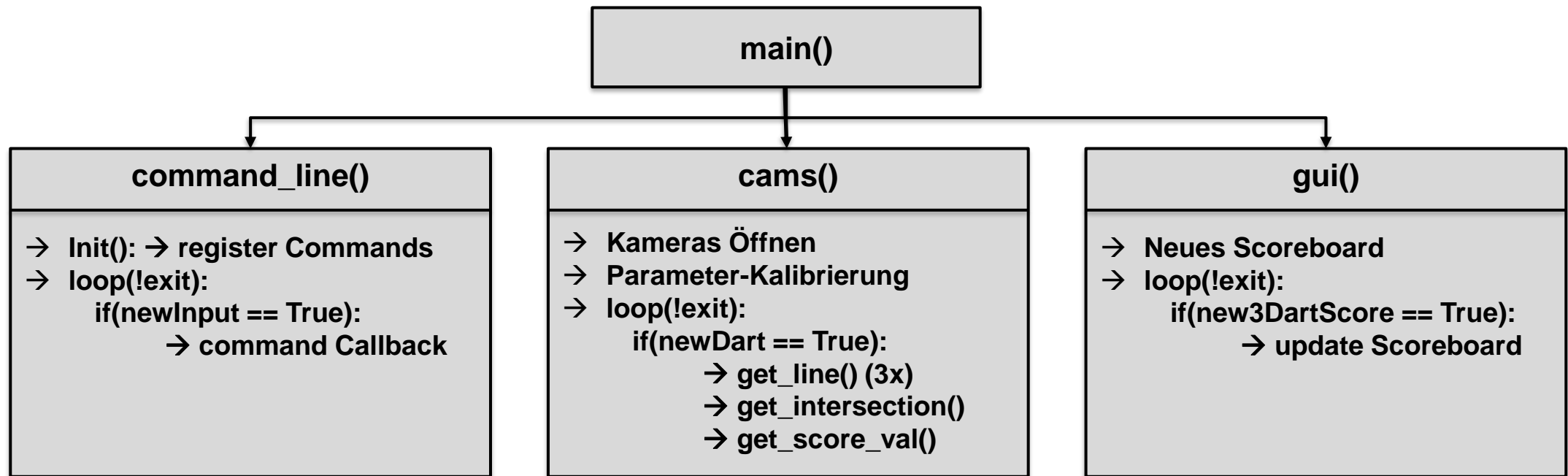
- Wichtig: Korrektur der Kameraperspektive
 - Warp zum „idealen“ Kreis
- Idee: Schnittpunktberechnung der 3 Perspektiven
 - Dart im Board finden und isolieren
 - Eine Gerade längs durch den Pfeil und die Spitze ziehen
 - Schnittpunkte aller 3 Gerade berechnen
 - Schnittpunkt zurückrechnen in Segment der Scheibe

RIGHT (warped)



KOMPONENTEN UND STRUKTUR DER SOFTWARE

- Aufteilung in 3 Threads:
 - Command Line
 - Scoreboard (GUI)
 - Kameras (Bildverarbeitung)



KOMPONENTEN UND STRUKTUR DER SOFTWARE

DARTS DETECTION

1. Erkennung eines neuen Darts:

- Vergleich letztes Bild mit aktuellem Bild
 - Rauschreduktion, Kalibrierung, Differenzbild, Schärfung, Kantenbild, Binärbild
 - Vergleich Pixelsumme mit Mindestschwellwert → Differenzstatus zurückgeben

2. Berechnung der Polarkoordinaten der Linie:

- Rauschreduktion, Kalibrierung, Differenzbild, Schärfung, Kantenbild, Binärbild
- Barrel / Shaft Erkennung: Konturen finden → Rechteck um Kontur legen
 - Nur Rechtecke um schmale, längliche Konturen (→ Aspekt-Verhältnis) mit Mindestfläche zeichnen
 - Wenn keine Kontur die Bedingung erfüllt, weiter mit Binärbild
- Kantenerkennung durch Hough-Transformation (2 Maxima → Mittelwertbild)
- Rückgabe der Polarkoordinaten der Linie durch den Dart

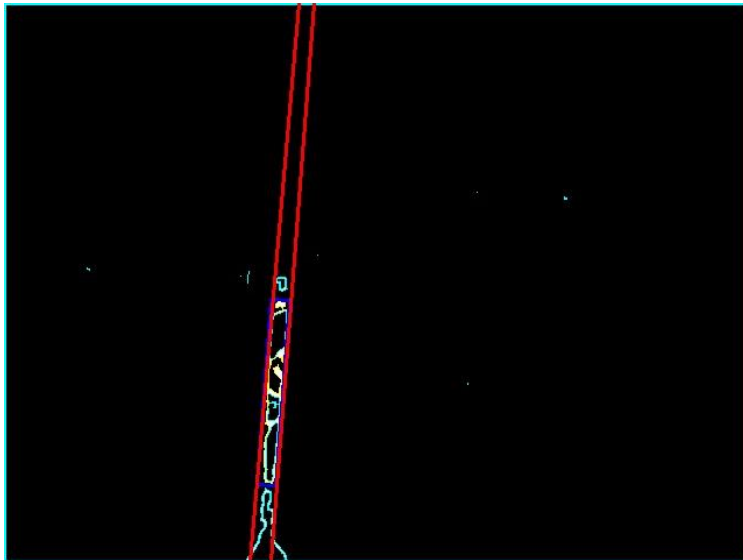
3. Schnittpunktberechnung der 3 Linien (grafisch oder mathematisch) & Rückrechnung des Pixels in Sektor

4. Nach 3 Pfeilen → warten bis Differenz zwischen aktuellem Bild und leerer Scheibe kleiner Schwellwert

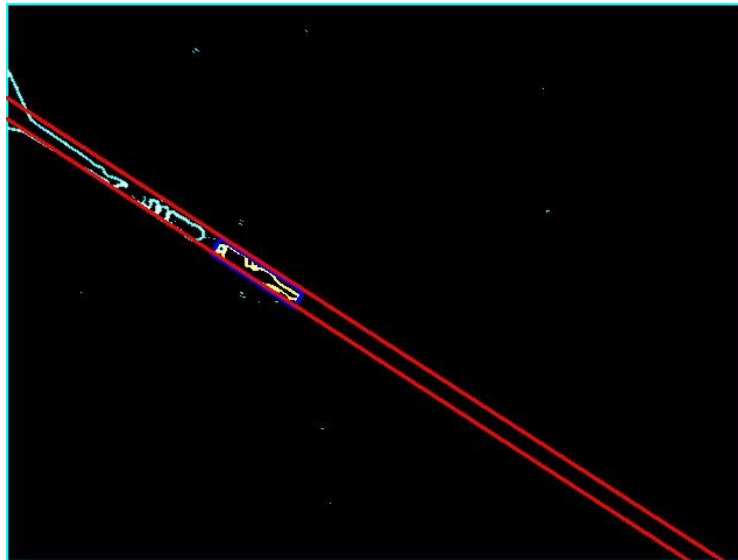
6

ERGEBNISSE

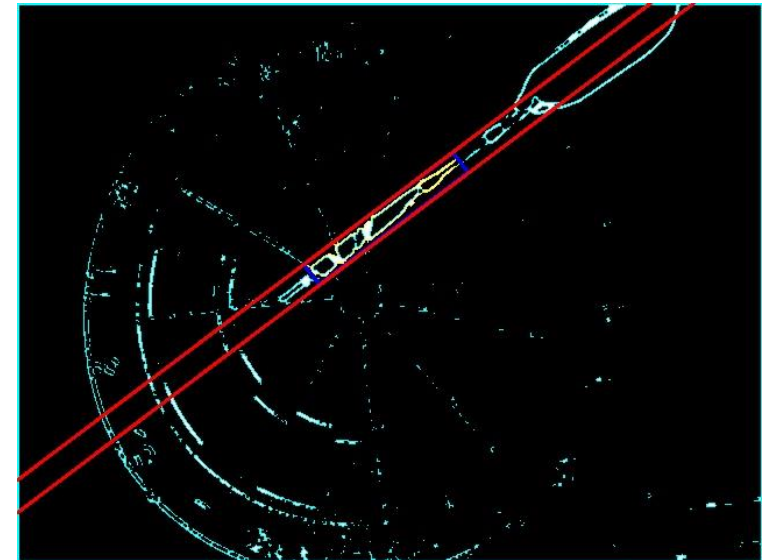
TOP



RIGHT

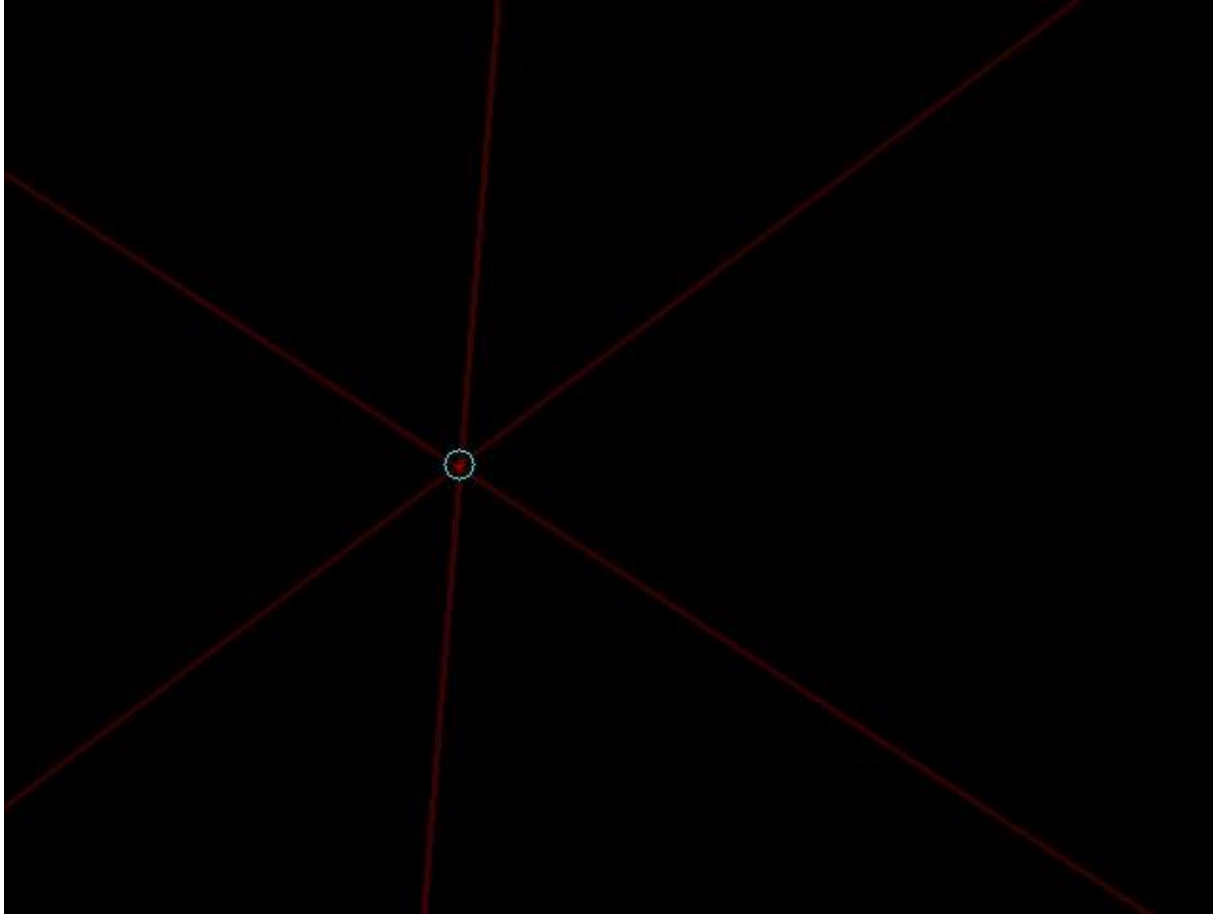


LEFT



ERGEBNISSE

Schnittpunkt



Command Line

```
C:\Users\xsale\Dokumente\HAWH\WiSe2024\Bildverarbeitung\Darts\darts\src\c++\darts\x64\Release\darts.exe
type 'exit' to quit
Command Line is active
Welcome to the Darts Command Line

Dart is (String): Single 11
Dart is (int Val): 11
```

Original (Top)



ERGEBNISSE

Scoreboard vor den ersten 3 Darts

Darts Scoreboard

player	set	leg	score	last throw
Lukas	0	0	501	0
Mika	0	0	501	0
Marek	0	0	501	0

press [Enter] to start the game
Leg 0 and it's Lukas to throw first
Game On!

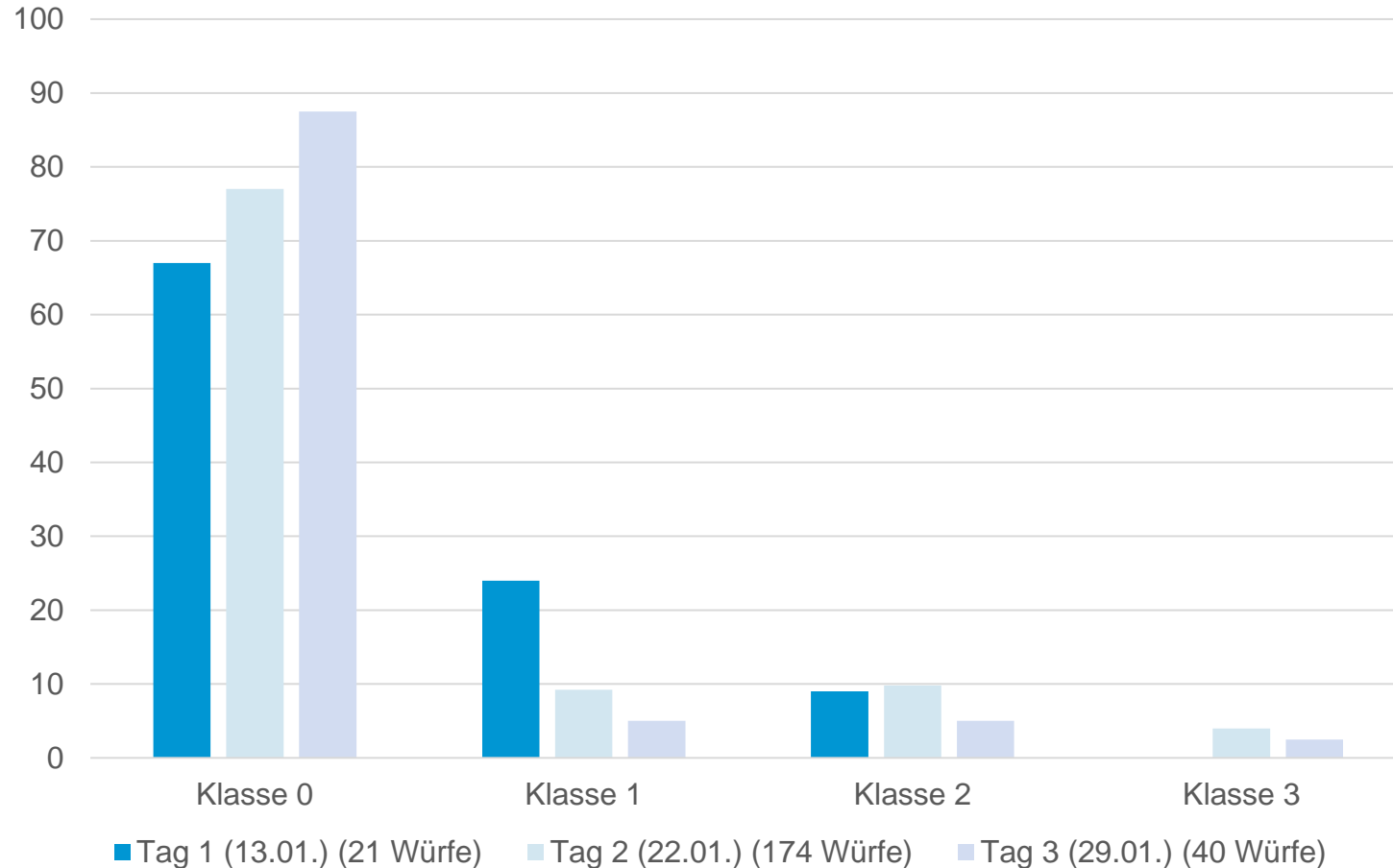
Scoreboard nach den ersten 3 Darts

Darts Scoreboard

player	set	leg	score	last throw
Lukas	0	0	490	11
Mika	0	0	501	0
Marek	0	0	501	0

ERGEBNISSE

- Statistische Auswertung: an drei Tagen → zwischen den Tagen Änderungen (Verbesserung) vorgenommen
 - Einfarbige Flights minimieren das Finden von Konturen außerhalb Barrel / Shaft (Tag 3)



Legende Klassen:

0 := Richtig erkannt

1 := Sektor richtig S/D/T falsch

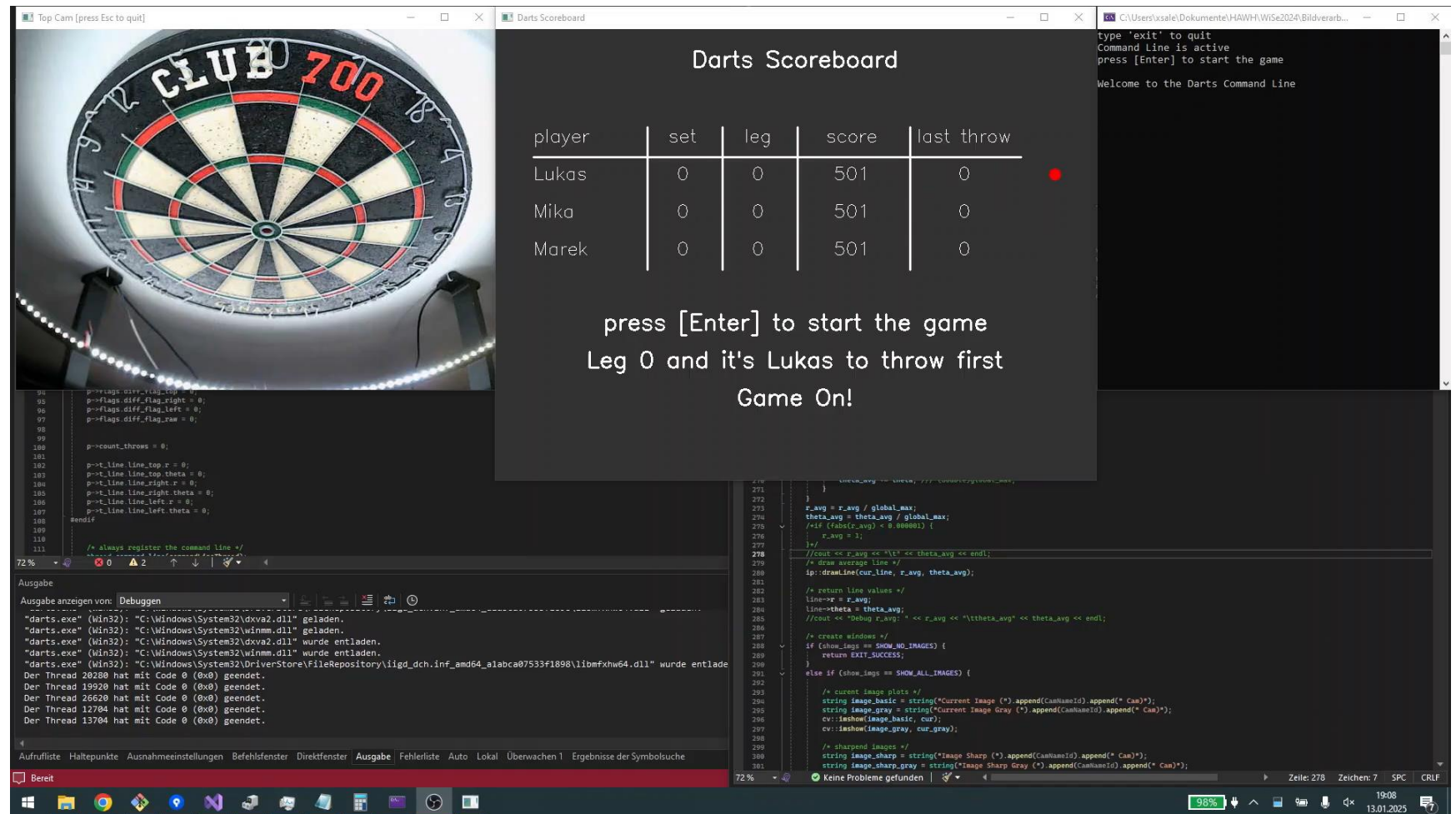
2 := Nachbarsektor erkannt

3 := komplett falsch

→ Balken zeigen Anteil der Klasse des Tages in [%]

ERGEBNISSE

➤ Video:



The screenshot displays a darts game interface with three main components:

- Dartboard:** A standard dartboard with the text "CLUB 200" at the top.
- Darts Scoreboard:** A table showing the current scores for three players: Lukas, Mika, and Marek. Each player has a score of 501 in the 'score' column and 0 in the 'last throw' column. The 'set' and 'leg' columns are also 0.
- Command Line:** A text area with instructions: "type 'exit' to quit", "Command Line is active", "press [Enter] to start the game", and "Welcome to the Darts Command Line".

Below the scoreboard, there is a text prompt: "press [Enter] to start the game", "Leg 0 and it's Lukas to throw first", and "Game On!".

The interface also includes a code editor at the bottom with C++ code for image processing and a debug console showing the execution of the program.

ERGEBNISSE

➤ Video:

```
Dart is (int Val): 12
Dart is (String): Single 9
Dart is (int Val): 9
removing darts ...
ready ...
Dart is (String): Triple 20
Dart is (int Val): 60
Dart is (String): Single 4
Dart is (int Val): 4
Dart is (String): Out of Board
Dart is (int Val): 0
removing darts ...
ready ...
Dart is (String): Single 9
Dart is (int Val): 9
Dart is (String): Single 20
Dart is (int Val): 20
Dart is (String): Single 7
Dart is (int Val): 7
removing darts ...
ready ...
Dart is (String): Out of Board
Dart is (int Val): 0
Dart is (String): Out of Board
Dart is (int Val): 0
Dart is (String): Out of Board
Dart is (int Val): 0
removing darts ...
ready ...
```

Darts Scoreboard

player	set	leg	score	last throw
Lukas	0	0	432	36
Mika	0	0	437	0

SCHWIERIGKEITEN

- Allgemein:
 - Licht (→ veränderte Parameter)
 - Kontur der Darts (z. B. Ringe am Barrel, rundes Barrel)
 - Unterschiedliche Farben der Darts
 - Schwarze Spitzen, schwarzes Feld
 - Dart kaum in der Kamera
 - Zahlreiche Positionen und Winkel der Darts
 - Millimeter „dicke“ Drähte ($< 0,5 \text{ mm}$)
 - Fisheye verfälschte Radien
- Kantenerkennung in Differenzbildern
 - Rechenintensiv (noch i. O.)
 - Kanten parallel zur Spitze im gleichen Abstand
 - Deutlich Verbessert durch Konturen (Barrel Detection)
 - **Veränderung des vorherigen Darts**



SOFTWARE

- Visual Studio 2022
- [2] Salewski, Mika Paul: <https://github.com/mpsalewski/> (noch nicht veröffentlicht)

REFERENZEN

- [1] Hensel, Marc: Workshop Bildverarbeitung mit OpenCV und C++, 2024
- [2] Salewski, Mika Paul: <https://github.com/mpsalewski/> (noch nicht veröffentlicht)