

Ziel meines gewählten Projektes ist es,
die Roboter, die Nao's mit Ihren Armen in bestimmte Richtungen oder auf Objekte deuten zu lassen.
Der Grund für die Implementierung dieser Funktion ist, dass das deuten auf bestimmte Objekte
Später zum Testen genutzt werden kann z.B.

Ready Position Korrekt
Default Position Korrekt
Ball richtig erkannt

Im ersten Schritt meines Projektes soll umgesetzt werden:

Ein Roboter soll auf das eigene Tor zeigen können.
Erstellen einer Scene mit einem Spieler → 1vsDummies.rsi2 config/Scenes

Stand der Dinge:

Entsprechende Ansätze sind vorhanden (PointAt).
Diese soll es ermöglichen die Nao's auf bestimmte relative Koordinaten zeigen zu lassen.

Vorhandene Ansätze:

- Interessante Skills aus Skills.h:

```
/**
 * ACTION SKILL
 * This skill lets one arm point at some point.
 * @param localPoint The point in robot-relative coordinates
 */
SKILL_INTERFACE(PointAt, (const Vector3f&) localPoint);

/**
 * ACTION SKILL
 * This skill lets a specific arm point at some point.
 * @param localPoint The point in robot-relative coordinates
 * @param arm The arm that shall be used for pointing
 */
SKILL_INTERFACE(PointAtWithArm, (const Vector3f&) localPoint, (Arms::Arm) arm);
```

- Vielleicht weitere nützliche Skills:

```
/**
 * ACTION SKILL
 * This skill moves the head so that a camera looks at specified angles.
 * Note: Is not dynamic. If you want the camera to be dynamic, you must change the input angles
 * @param pan The target pan angle
 * @param tilt The target tilt angle
 * @param speed The speed with which to move the head, in radians/sec
 * @param camera The camera which should have the specified angles
 * @param calibrationMode Whether to set the mode to calibrationMode instead of panAndTiltMode,
 * which disables clipping and interpolation of angles.
 */
SKILL_INTERFACE(LookAtAngles, (Angle) pan, (Angle) tilt, (float)(180_deg) speed,
  (HeadMotionRequest::CameraControlMode)(HeadMotionRequest::autoCamera) camera,
  (bool)(false) calibrationMode);

/**
 * ACTION SKILL
 * This skill turns the robot to look forward at a target.
 * Does not move head
 * @param target The position in robot relative coordinates to turn to
 * @param margin The tolerance for the skill to be done
 */
SKILL_INTERFACE(TurnToPoint, (const Vector2f&) target, (Angle)(5_deg) margin);
```

Technische Umsetzung:

- Verwenden der PointAt Skills
- Zeit auslesen für Bestimmung der Dauer des Zeigens
 - game time
 - Unix System Zeit

Ablaufplan:

- relativen Koordinaten der Roboter verstehen, z.B. Wo ist Nullpunkt → Anstoßpunkt
- Testen der vorhandenen Skills.
- Vergangene Zeit Messen (Zeit Auslesen)
- Testen im Simulator da hier einfacher Debug Infos ausgelesen werden können
- Eventuelle Live-Tests