| Abend | **Lernziel** | **Thema / Inhalt** | **Methode** | **Zeitbedarf** | **Hausaufgaben** |
| --- | --- | --- | --- | --- | --- |
| 1. Abend | Konzept Objektorientiertes programmieren anhand von G-Blocks verstehen und eine einfache Steuerung implementieren | **Vorstellung (Wer bin ich? Problem-Based Learning)**  **EVA – Prinzip (Sensoren – Verarbeitung – Aktoren)**  **Installieren der Entwicklungsumgebung**  **Aufgabe 1 (GoAndReturn)**   * 2s gerade aus * Sound abspielen * Pixelgrafik anzeigen (Bild- * 180° kehren * 2s zurück   *Verwendetes Material:*   * *Sequenz von Objekten, statische Properties* * *Die 4 verschiedenen Motorklassen und ihre Properties* * *Bildschirm-Klasse (Bild-Bearbeiten, Pixelgrafik)* * *Sound-Klasse (Ton-Bearbeiten, MP3 konvertieren)* * *Timer-Klasse* | Vortrag  Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben  Verschiedene Varianten selber austesten | 20‘  15‘  15‘  120‘ |  |
| 2. Abend | Ablaufstrukturen im Programm gezielt und richtig verwenden | **Aufgabe 2 (Sirene)**   * Wechsel zwischen C# and G# (0.5s Intervall) * Synchrones Blinken (rot) mit dem Tonwechsel * Solange Taster gedrückt ON sonst OFF   *Verwendetes Material:*   * *Sound-Klasse (Ton abspeilen)* * *LED-Klasse* * *Timer-Klasse mit Taster Event* * *Loop Block* * *Multithreading*   **Aufgabe 3 (Staubsauger Roboter)**   * Fahrzeug so programmieren, dass es Hindernissen ausweicht  1. Smiley traurig, wenn Taster gedrückt sonst lachend 2. Wenn Smiley traurig, Fz gerade aus, sonst Motoren stopp 3. Anstelle Taster, Distanz-Sensor verwenden 4. Anstelle stoppen, ausweichen  * Wettkampf in Arena   *Verwendetes Material:*   * *Verzweigung innerhalb Infinit-Loop (Grundstruktur für Steuerung)* * *Verzweigung durch Touch- und Distanz-Sensor steuern* | Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben    Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben | 60‘  140‘ |  |
| 3. Abend | Property-Binding richtig anwenden | **Aufgabe 4 (Bremskurve)**  Fahrzeug so programmieren, dass es mit 100% Leistung gerade aus fährt bis ein Hindernis näher als 50cm auftaucht. Dann abbremsen, so dass das Fz 30cm vor dem Hindernis stehen bleibt.   1. Lösungsversuch mit verschachtelten if-then-else 2. Lösung mit linearer Bremskurve    1. Mathematische Herleitung der Distanz-Leistungskurve    2. Implementation des mathematischen Modell    3. Testen der Lösung   *Verwendetes Material:*   * *Mathe-Klasse* * *Datentypes* * *Lineare Funktion (Math. Herleitung)* | Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben | 60‘  140‘ | Übungen lineare Funktion |
| 4. Abend | Eigener Block (Klasse) kreieren | **Aufgabe 4a (Bremskurve, CHIP-tuning)**  Das Programm so erweitern, dass die Parameter (Steigung, y-Achsenabschnitt) aus den beiden Stützpunkten vom Steuerungsprogramm berechnet werden.  Eigener Block **LineareFunktion** kreieren und in Applikation verwenden und testen  *Verwendetes Material:*   * *Variablen definieren / schreiben / lesen* * *Datentypes* * *Eigeneblocks*   + *Interface design*   + *Implementation* | Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben | 90‘  110‘ |  |
| 5. Abend | Re-Use des eigenen Blocks | **Aufgabe 5 (Gaspedal und Lenkrad)**  Das Fahrzeug über eine kabelgebundene RC steuern. Dabei werden zwei Drehregler verwendet. Die jeweiligen Drehwinkel werden mit LinearenFunktion-Objekten in **Steuerung** und **Leistung** umgerechnet. Mit Mittlerer-Taste kann Not-Stop (Leistung und Steuerung 0%) gemacht werden. Weiter sollen die Parameter auf der Anzeige angezeigt werden. Falls Rückwärts gefahren wird, Warnhupe und Warnblinker ON  *Verwendetes Material:*   * *Motoren als Drehregler* * *Stein-Tasten Klasse* * *Text-Ausgabe* * *String-Conncationation* | Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben | 200‘ |  |
| 6. Abend | Bluetooth (BT) Nachrichten senden (aufbereiten) und empfangen (verarbeiten) können | **Aufgabe 6 (BT Connection)**  Pairen Sie zwei EV3 und senden die verschiedenen Stein-Tasten als String zum Empfänger. Der Empfänger zeigt die Strings an.  Erweitern Sie das Programm, so dass auch UP, DOWN,…. vom Empfänger gesagt wird.  Senden Sie einen Zahlenwert (vom Drehregler, Lagesensor,…) und zeigen Sie diesen Wert auf der Empfangsseite an.  Erweitern Sie das Programm so, dass 2 Zahlenwerte und die Steintaste gesendet werden können und auf der Empfängerseite angezeigt werden.  *Verwendetes Material:*   * *Bluetooth Meldungen senden und empfangen* * *Case-Switch Statement* | Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben | 60‘  40‘  40‘  60‘ |  |
| 7. Abend | Anwendung im Team (Schnittstellen) entwickeln können | **Aufgabe 7 (RC-Car)**  Splitten Sie das Programm von Aufgabe 5 in zwei Teile:   1. Fz empfängt Lenkung und Leistung über BT 2. Eine Remote-Conrol (RC) sendet die Leistung und Lenkungsdaten via BT an Fz. Die Parameter werden über zwei Drehregler an der RC berechnet (mit lin. Funktion für die Empfindlichkeit). Die Lekung und Leistungsdaten werden auf dem Display der RC angezeigt.   Testen und Debugen, Refactoring, Cleancode  Erweitern Sie beide Teile wie folgt:   1. Fz verhindert selbst dass es irgendwo nach vorne einschiessen kann. 2. RC kann die Parameter ebenfalls über Lagesensoren berechnen und senden. Die Schnittstelle darf nicht ändern (Das Fz merkt nicht, von welchen Sensoren die Lenkung und Leistung kommen)   Testen und Debugen, Refactoring, Cleancode  *Verwendetes Material:*   * *Test-Driven approach* * *Cleancode Regeln* * *Schnittstellenvertrag* * *SW Entwicklung im Team* | Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben | 100‘  100‘ |  |
| 8. Abend | Anwendung im Team (Schnittstellen) entwickeln können | **Aufgabe 8 (Linienfolger digital)**  Programmieren Sie das Fz so, dass es einer **zweifarbigen** Linie (F1-Strecke Monza) folgen kann. Sobald das Fz die Startlinie überfährt, beginnt die Zeit zu laufen (wird angezeigt auf dem Display) und stoppet sobald die Ziellinie überfahren wird.  *Verwendetes Material:*   * *Farb-Sensor* * *Timer-Block (Stop-Uhr)* | Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben | 200‘ |  |
| 9. Abend | Anwendung im Team (Schnittstellen) entwickeln können | **Aufgabe 9 (Linienfolger analog)**  Programmieren Sie das Fz so, dass es einer **einfarbigen** Linie (F1-Strecke Monza) folgen kann. Sobald das Fz die Startlinie überfährt, beginnt die Zeit zu laufen (wird angezeigt auf dem Display) und stoppet sobald die Ziellinie überfahren wird.  *Verwendetes Material:*   * *Helligkeits-Sensor* | Probieren, Vormachen, Nachmachen mit theoretischen kurzen Einschüben | 200‘ |  |

|  |  |
| --- | --- |
| **Bemerkungen:** | * Jeder Abend dauert 4 Lektionen. * Der Unterrichtsplan kann bei Bedarf dem vorhandenen Wissen der Klasse angepasst werden. * Die Studierenden lösen die Übungen auf ihren privaten Notebooks (Entwicklungsumgebung für Windows wird zur Verfügung gestellt). * Aufgabe 7 wird im 2-er Team bearbeitet und als Leistungsnachweis abgegeben. Darüber wird mit dem Dozenten ein Fachgespräch geführt und bewertet (Erfahrungsnote). * Die Modullernzielkontrolle (MLZ) findet am 8.Abend in Form einer Moodle Prüfung (Open Books) statt. |