## 01\_Infoblatt Lehrperson Informatik

## 

**Allgemeine Hinweise**

Arbeitsform: Die folgenden Übungen werden an der Wandtafel oder am Whiteboard von der LP während dem Unterricht aufgezeichnet und aufgeschrieben. Dies ermöglicht ein schrittweises additives Vorgehen und somit den Einbezug der Lernenden in Form des Aktivierens durch Fragen und Aufforderung zum Mitdenken. Zudem ist es sehr vorteilhaft für die Lernenden bei der Entwicklung der Programme die verschiedenen Sequenzen nebeneinander zu sehen. Denn es wird nötig sein immer wieder auf ältere Sequenzen zurückzugreifen um sie ergänzen/weiter zu entwickeln.

Die Unterrichtseinheit besteht aus 4 Themen

1. **Begriffe und Definitionen** zu Algorythmus und Flussdiagramm werden eingeführt
2. **Simulation** zu kürzestem Weg/Problem des Handlungsreisenden wird vorgeführt und an die Programmierlogik dahinter herangeführt.
3. **Erste Programmier-Übungen auf PGLU**, die sich schon konkret auf die Programmierung des Ameisenroboters beziehen
4. **Braitenberg-Vehikel,** eine Inspiration für die Erzeugung lebendig wirkender Maschinen

**Einleitende Worte:**

“Sie haben Anfangs des Projektes ja schon etwas dazu gehört, wie Ameisen als Vorlage für mathematische und informatische Fragestellungen genutzt werden können. In folgenden Unterricht nehmen wir die Ameisenalgorythmen noch einmal kurz auf und wollen uns dann aber gezielt dem Programmieren eigener Ameisenverhalten widmen. Hierzu verwenden wir die Programmieroberfläche der PGLU-Einheit. Diese ermöglicht einen vereinfachten Zugang zum Programmieren und ist auf die Arduino-Platinen abgestimmt ist, die beim Ameisenroboter den Sie später bauen zum Einsatz kommen.“

“Da wir das Programmieren praktisch von 0 aus aufbauen, ist mir wichtig, dass sich diejenigen unter Ihnen melden, die schon über ein breites Vorwissen beim Programmieren verfügen. Es soll Ihnen nicht langweilig werden und ich kann ihnen verschiedene Angebote für die weitere Partizipation machen“

Für Lernende mit fortgeschrittenem Wissen im Feld der Informatik wird empfohlen, den 1. Teil der Einführung mit zu verfolgen und hierbei den Fokus auf die Vermittlung informatischer Grundkenntnisse zu legen und zum Anderen um die PGLU-Programmierplattform kennenlernen.

In einem 2. Teil stelle ich Ihnen gerne frei sich selbstständig mit den folgenden Punkten auseinander zu setzen:

-Befehle hinter der Blocksprache betrachten

-Aufgaben in Code programmieren und in die PGLU-Programmierplattform einfügen

-Codes hinter der Simulation betrachten die sie in Kürze vorgeführt bekommen. “

**1. Begriffe und Definitionen**

Frage: Was ist ein Algorythmus?

Antwort : Ein Algorythmus ist ein schrittweises präzises Verfahren zur Lösung eines Problems.

Was das bedeutet und wie ein Algorythmus notiert wird lernen sie im folgenden Beispiel kennen. Hierzu zunächst einige Grundbegriffe und Notierweisen: **Auf Whiteboard aufzeichnen – muss nicht von den Studierenden notiert werden / wird stehen gelassen**

|  |  |
| --- | --- |
| Algorythmus : | Name (allfällige Parameter) |
| Beginn: |  |
| Ende: |  |
| Eingangswert: |  |
| Ausgangswert: |  |
| Anweisung: | textuell |
| Sequenz: | Eine Abfolge von Anweisungen / Befehlen |
| Verzweigung / Frage: |  |
| Wiederholung: |  |

**Übung Notation**

*Hinweis für LP: Bis alle Grundoperationen eingeführt sind wird schrittweise aufgezeigt wie das Ganze aufzuzeichnen und zu erklären ist um das schrittweise Vorgehen/Vermitteln anschaulich zu machen. In Klammern befindet sich jeweils der gesprochene Text.*

**“**Um Sie mit der Logik und Notation vertraut zu machen möchte ich mit Ihnen einen Algorythmus erarbeiten, den Sie möglicherweise alle schon einmal angewendet haben“:





“Starten wir nun also mit dem Programm“:

|  |  |
| --- | --- |
|  | Frage |
|  | 2 mögliche Antworten |
|  | Weiteres Vorgehen  -**Bei Lernenden erfragen** |
|  | Bedingungen erfüllt ?  **Bei Lernenden erfragen**  Ja, weiter  Nein, Schleife |
|  | Weiteres Vorgehen und Programmende |

Die LP erklärt:

“Hiermit haben wir auch schon die Grundoperationen angewendet auf welchen praktisch alle Programme beruhen. Dies sind die Sequenz, Selektion und Iteration.

|  |  |
| --- | --- |
| Die Sequenz beschreibt hintereinander auszuführende Programmanweisungen  **Frage an die Lernenden: Wo kam dies beim Pizzateigprogramm vor?**  ->Auf Elemente zeigen oder farblich hervorheben, die im Bild rechts aufgezeigt sind |  |
| Die Auswahl oder Selektion beschreibt eine Verzweigung im Programm  **Frage an die Lernenden: Wo kam dies beim Pizzateigprogramm vor?**  ->Auf Elemente zeigen oder farblich hervorheben, die im Bild rechts aufgezeigt sind |  |
| Die Iteration oder Wiederholung beschreibt Schleifen die in das Programm eingebaut werden  **Frage an die Lernenden: Wo kam dies beim Pizzateigprogramm vor?**  ->Auf Elemente zeigen oder farblich hervorheben, die im Bild rechts aufgezeigt sind |  |

**2. Simulation “Ameisenwelt“**

Ausser dem Herstellen von Pizzateig kann man natürlich noch andere Programme schreiben. Für das Bionikprojekt hat Prof. Gianni die Pietro, Dozent beim Institut LifeSciences an der FHNW, ein Programm zur Veranschaulichung des Ameisenalgorythmus zum Problem des Handlungsreisenden geschrieben. Es geht darum, dass die Ameisen eine Futterquelle suchen und später auch finden und (mit der Zeit) den kürzesten Weg davon zurück finden.

Zeigen der Simulation 🡪 AmeisenWelt.iml

Zur besseren Verständlichkeit wird die Herangehensweise an den Aufbau des Programmes in 4 Szenarien unterteilt. Es wird versucht ein natürliches Verhalten von Ameisen in einem Programm nach zu bilden – diese Vorgehensweise nennt sich im Fachjargon -Reverse Ingeniering-. Das Programm wurde als Java-App geschrieben.

Der Ausgangspunkt ist ein Nest, aus dem Ameisen kommen und auf Futtersuche gehen

**Szenario ohne Futter**

**Sequenz 1:**

|  |  |
| --- | --- |
| 1a) |  |
| 1b) |  |

**Sequenz 2:**

|  |  |
| --- | --- |
|  |  |

**Sequenz 3:**

|  |  |
| --- | --- |
|  |  |

**Sequenz 4:**

|  |  |
| --- | --- |
|  |  |

**Sequenz 5:**

|  |  |
| --- | --- |
|  |  |
| Das Ganze beschreibt das Programm einer Ameise- die Anzahl kann nun beliebig hochgefahren werden |  |

**Szenario 2 mit Futter**

Frage an die Studierenden: “ Welche Ablaufdiagramme ändern sich?“

Antwort: Strecke bis müde ( ), Strecke bis Nest( )

**Sequenz 6:**

|  |  |
| --- | --- |
| Das rot markierte wird bei Sequenz 2 / Strecke bis müde ( ) hinzugefügt |  |

**Sequenz 7:**

|  |  |
| --- | --- |
| **Die Lernenden sollen selbstständig versuchen zu ergänzen**  Das rot markierte wird bei Sequenz 4 / Zyklus ( ) hinzugefügt |  |

**Sequenz 8:**

|  |  |
| --- | --- |
| **Die Lernenden sollen selbstständig versuchen zu ergänzen**  Das rot markierte wird bei Sequenz 4 / Zyklus ( ) hinzugefügt |  |

**Szenario 3 mit Pheromonspur**

Frage an die Studierenden “Die Ameisen legen nun eine Pheromonspur: “ Welche Ablaufdiagramme ändern sich?“

Antwort: Strecke zurück zum Nest( )

**Sequenz 9:**

|  |  |
| --- | --- |
| Das rot markierte wird bei Sequenz 3 / Strecke bis Nest ( ) hinzugefügt |  |

**Szenario 4 Pheromonspur folgen**

Frage an die Studierenden “Die Ameisen folgen nun einer Pheromonspur: “ Welche Ablaufdiagramme ändern sich?“

Antwort: Schritt( )

**Sequenz 10:**

|  |  |
| --- | --- |
| Das rot markierte wird bei Sequenz 1a / Schritt ( ) hinzugefügt |  |

1. **Erste Programmier-Übungen auf PGLU**

**Anlage des Roboters**

|  |  |
| --- | --- |
| Der Ameisenroboter den Sie bauen werden wird über folgende Elemente verfügen, die sie über die Programmierung ansteuern können.  S1, S2: Lichtsensoren  S3, S4: Tastsensoren  M1, M2: Motoren für die Beine  L1-L4: LED-Ports die verwendet werden können |  |

**Programmieren auf PGLU**

|  |  |
| --- | --- |
| Um die Elemente ansteuern zu können brauchen wir eine entsprechende Programmieroberfläche.  Öffnen Sie hierzu ihren Rechner und vergewissern Sie sich, dass Sie mit dem Internet verbunden sind.  Tippen Sie in ihrem Browser folgende Internetadresse ein:  <https://pglu.ch>  Gehen Sie auf den Reiter «Editor» und auf «Editor online» im Klappmenü  In der Adresszeile wird nun «mach.pglu.ch“geöffnet und sie sehen nun die Programmierplattform |  |

**Programmieren einer Sequenz:**

|  |  |
| --- | --- |
| “Sie möchten, dass sich der Roboter vorwärts bewegt. Wie sieht das in einem Flussdiagramm aus?“ (ev. kurze Diskussion unter den Lernenden ) |  |
| * “Wir beginnen mit dem Motor, nehmen 2mal den Block Motor und setzen die entsprechenden Nummern, die Leistung wird standardmässig mit 100% ausgegeben” * LP demonstriert an ihrem Rechner, wie die Sequenz programmiert wird. Nachvollzug durch die Studierenden auf ihrem Rechner * Test durch Simulation |  |
| * “Ihr Roboter bewegt sich nun unendlich lange nach vorne- wir möchten aber, dass er dies nur für 5 Sekunden macht und danach 5 Sekunden innehält. Wir fügen hierzu den Befehl Pause ein. Dieser besagt, dass das Programm die Sequenz ausführt und 5s keine anderen Befehle entgegennimmt. |  |
| * Wie würden Sie eine Kurve nach links programmieren?   Verhältnis zwischen den Motorenleistungen ergibt Kurvenradius. Wäre Motor auf 0 gestellt, ergäbe dies ein Drehen des Roboters fast an Ort |  |

**Programmieren einer Selektion:**

|  |  |
| --- | --- |
| Sie möchten,  “Dass wenn Sensor 1 aktiviert ist, die LED 1 leuchtet. Ansonsten leuchtet LED 2.“ (ev. kurze Diskussion unter den Lernenden ) |  |
| Sie möchten nun zusätzlich,  “Dass der Sensor 1 nur alle 2 Sekunden abgefragt wird und in diesen Pausen auch keine LEDs leuchten“ | Ein Bild, das Screenshot, Bildschirm, Monitor, Computer enthält.  Automatisch generierte Beschreibung |

**Programmieren einer Iteration:**

|  |  |
| --- | --- |
| Sie möchten,  “Dass die LED1 zehnmal hintereinander im Sekundentakt blinkt “ | Ein Bild, das Screenshot, Monitor, Bildschirm, sitzend enthält.  Automatisch generierte Beschreibung |

**(Programmieren mit Sequenzen)**

|  |  |
| --- | --- |
| Für die Übersichtlichkeit des Codes kann es von Vorteil sein Sequenzen ausserhalb des Hauptloopes zu beschreiben  Hier ein Beispiel hierzu | Ein Bild, das Screenshot, Monitor, Bildschirm, Fernsehen enthält.  Automatisch generierte Beschreibung |

**Selbständige Programmieraufgaben**

Jeweils eine Aufgabe, etwas Zeit zum Lösen, anschliessend gemeinsame Besprechung/Vorstellung

1. Aufgabe: Vehikel 10 Sek. geradeaus fahren und anschliessend anhalten lassen
2. Aufgabe: Vehikel fährt rückwärts
3. Aufgabe: Vehikel dreht sich im Uhrzeigersinn an Ort und Stelle
4. Aufgabe: Vehikel dreht sich um 180o (schwierigere Aufgabe)
5. Aufgabe: Vehikel fährt geradeaus bis es auf Hindernis stösst
6. Aufgabe: fährt bei Kollision rückwärts, wechselt Richtung
7. **Braitenberg-Vehikel**

Der Neurologe und Hirnforscher Valentino Braitenberg hat sich in einem Gedankenexperiment damit auseinandergesetzt welche Anforderungen ein Organismus oder eine Maschine erfüllen muss um lebensechtes Verhalten zeigen, respektive simulieren zu können. Ein erster Hinweis: es braucht erstaunlich wenig um sogar komplex wirkendes Verhalten zu simulieren. Ein zweiter Hinweis: mit der Programmierbarkeit ihrer Ameisenroboter könnten sie noch viel komplexere Verhaltensmuster entwerfen.

Die folgenden Beispiele aus seinem 1986 erschienen Buch *Künstliche Wesen*. Verhalten kybernetischer Vehikel1 sollen Sie inspirieren ein spannendes oder gar verblüffende Verhalten für ihren Ameisenroboter auszuhecken.

Zeigen Sie hierzu

02\_kybernetik-braitenberg-vehikel.pdf

1Valentin Braitenberg: *Künstliche Wesen*. Verhalten kybernetischer Vehikel. Friedr. Vieweg & Sohn, Braunschweig / Wiesbaden 1986,