**4 Plakate für die Schülerinnen und Schüler:**

|  |
| --- |
| **Ablaufschritte** |
| **Schritte mit einem PC/Notebook** | **Schritte mit einem Tablet** |
| Internetseite aufrufen: [www.calliope.cc](http://www.calliope.cc)  | Calliope – App öffnen   🡪 „Code erstellen“ wählen  |
| Programmieren --> Editoren --> Open Roberta LAB  | „Open Roberta NEPO“ wählen  |
| Benutzer --> anmelden --> neu --> Benutzername/Passwort  | Benutzer --> anmelden --> neu --> Benutzername/Passwort  |
| Bearbeiten --> speichern unter --> Programmnamen eingeben --> OK --> Code schreiben  | Bearbeiten --> speichern unter --> Programmnamen eingeben --> OK --> Code schreiben |
| Weiter Coden und immer wieder zwischendurch speichern.Jedem neuen Programm einen neuen Namen geben und abspeichern  | Weiter Coden und immer wieder zwischendurch speichern.Jedem neuen Programm einen neuen Namen geben und abspeichern |
| Simulation nutzen  | Simulation nutzen  |
| Code auf den Calliope mini übertragen und das Programm ausführen  | Code auf den Calliope mini übertragen und das Programm ausführen  |
| Schluss: Programm schließen--> Computer herunterfahren  | Schluss: Programm schließen--> App schließen  |
|  |
| **Code abspeichern und wieder aufrufen**  |
| **Im Browser:** | **In der Calliope App:** |
| Ein Bild, das Text enthält.  Automatisch generierte Beschreibung |  |
| **Code auf den Calliope mini übertragen** |
| **Im Browser** | **In der Calliope App** |
|  | Ein Bild, das Tisch enthält.  Automatisch generierte Beschreibung |

|  |
| --- |
| **Coding-Begriffe** |
| **Coden** | Ein Computerprogramm programmieren |  |
| **EVA-Prinzip** | * **Ei**ngabe: Ein Befehl wird eingegeben. (meist über Sensoren)
* **V**erarbeitung: Der Befehl wird im Prozessor des Computers verarbeitet und in passende Signale an einen Aktor übersetzt.)
* **A**usgabe (Ein Aktor setzt die Befehle um.)
 |
| **Editor** | Benutzeroberfläche, mit der man programmieren kann. (z.B. open Roberta NEPO) |  |
| **Sensor** | Das technische Gerät, welches einen Befehl auslöst (Tasten, Lichtsensor, Bewegungssensor, Ultraschallsensor für Entfernungen, Berührungssensor, …) |  |
| **Aktor** | Das technische Element, welches einen Befehl ausführt. (Licht: z.B. LED, Ton: z.B. Lautsprecher, Bewegung, z.B. Motor, …) |  |
| **Befehl** | Eindeutige Anweisung, die vom Computer ausgeführt wird. Ein Code ist in der Regel aus mehreren Befehlen zusammengesetzt. | C:\Users\gp\Pictures\Screenshots\Screenshot (752).png |
| **Sequenz** | Eine Folge von Befehlen*Zusatzinfo: „x ms warten“ bedeutet hier: den Befehl für x ms* ***ausführen/ halten*** | C:\Users\gp\Pictures\Screenshots\Screenshot (753).png |
| **Schleife** | Sich x-fach wiederholende Sequenz durch einen Wiederholungs-Befehl  | C:\Users\gp\Pictures\Screenshots\Screenshot (759).png |
| **Endlos-Schleife**  | Schleife einer sich unendlich wiederholenden Sequenz (bis zu ihrem eventuellen Abbruch) | C:\Users\gp\Pictures\Screenshots\Screenshot (760).png |
| **Bug** | Fehler im Programm |  |

|  |
| --- |
| **Hinweise zum Coden** |
| **Verschiedene mögliche Zugänge zum Editor NEPO** | Wähle [www.calliope.cc](http://www.calliope.cc) **oder** https:// lab.open-org, um den Editor zu öffnen. |  |
| **Code-Datei vor Verlust absichern** | Systemabsturz-Folgen vorbeugen: Zuerst Konto (er)öffnen, Programm Namen geben und speichern… immer wieder **zwischenspeichern**  | C:\Users\gp\Pictures\Screenshots\Screenshot (797).png |
| **Funktionieren von Codes mit Bedingungen/ Sensoren sicherstellen** | Der Befehl „***Wiederhole unendlich oft“*** ist zum „Einkleiden“ von Bedingungen/Sensoren (im Block „wenn, mache“ immer notwendig, damit der Code funktioniert.Erklärung: der Code wird in Bruchteilen von Sekunden durchlaufen. Er bliebe ohne Wiederholungsschleife als Sequenz unten hängen. Auf den Bedingungsblock mit Zugriff auf den Sensor kann dann nicht zugegriffen werden. | C:\Users\gp\Pictures\Screenshots\Screenshot (762).png |
| **Sequenzen** | Eine Sequenz läuft (vollständig, ohne Unterbrechung) von oben bis ganz nach unten durch und endet dann.D.h. eine laufende Sequenz wird z.B. durch andere Bedingungen nicht unterbrochen. | C:\Users\gp\Pictures\Screenshots\Screenshot (810).png |
| Einen Befehl für eine Zeitlang bis zum nächsten Befehl **halten/ausführen**, um Ausgaben für eine definierte Zeit sichtbar zu machen. | „Warte xy “ (= Millisekunden) | C:\Users\gp\Pictures\Screenshots\Screenshot (781).png |
| **Funktionieren des Simulators sicherstellen** | Den Simulator nach jedem Programmierschritt erneut **aus- und wieder einschalten**, damit er den jeweils letzten Programmierschritt zeigt. | C:\Users\gp\Pictures\Screenshots\Screenshot (803).png | 🡪 | C:\Users\gp\Pictures\Screenshots\Screenshot (804).png |