**Simon’s Brainfog Game V3.1**

**Aansluitingen Arduino**

**Afbeelding met tekst, Elektronische engineering, elektronica, stroomkring

Automatisch gegenereerde beschrijving**

**D2 Led 1 OUTPUT Kleurcode geel  
D3 Led 2 OUTPUT Kleurcode blauw  
D4 Led 3 OUTPUT Kleurcode groen  
D5 Led 4 OUTPUT Kleurcode rood  
D6 S1 INPUT Kleurcode geel  
D7 S2 INPUT Kleurcode blauw  
D8 S3 INPUT Kleurcode groen  
D9 S4 INPUT Kleurcode rood**

**11 Piezo SPKR OUTPUT Kleurcode oranje**

**A4 - SDA LCD DISPLAY DATA Kleurcode geel**

**A5 - SCL LCD DISPLAY CLOCK Kleurcode blauw**

**Afbeelding met elektronica, Elektronische engineering, Elektronisch onderdeel, Stroomkringonderdeel

Automatisch gegenereerde beschrijvingAansluitingen LCD-display (I2C)**

**In dit voorbeeld is een Adafruit 16x2 LCD monochrome display gebruikt met blauwe achtergrondverlichting en I2C aangesloten.**

**ANALYSE VAN HET SPEL**

**Initialisatie**:

Pinnen worden geïnitialiseerd voor input en output. Het LCD-scherm en de luidspreker worden geïnitialiseerd.

1. **Hoofdmenuweergave**: Het hoofdmenu wordt weergegeven op het LCD, en de gebruiker kan een moeilijkheidsniveau selecteren door de bijbehorende knoppen in te drukken.
2. **Knopfeedback**: Wanneer een moeilijkheidsniveau is geselecteerd, wordt een toon afgespeeld, licht de geselecteerde drukknop even op en wordt het geselecteerde moeilijkheidsniveau kort op het LCD-scherm weergegeven.
3. **Start van het spel**: Na het selecteren van een moeilijkheidsniveau begint het spel. Een bericht waarin de start van het spel wordt aangegeven, wordt op het LCD-scherm weergegeven.
4. **Genereren van het spel**: Er wordt een willekeurige volgorde van tonen gegenereerd op basis van het geselecteerde moeilijkheidsniveau.
5. **Spelverloop**: De Arduino speelt de gegenereerde volgorde van tonen af, en de speler wordt gevraagd om de volgorde te herhalen. Als de speler de volgorde correct herhaalt, gaat het spel verder; anders eindigt het spel.
6. **Aanpak van het einde van het spel**: Als de speler verliest, krijgen ze de optie om opnieuw te spelen. Als de speler wint, wordt hun uiteindelijke score berekend op basis van het aantal voltooide rondes en eventuele extra punten op basis van het moeilijkheidsniveau. De uiteindelijke score wordt weergegeven op het LCD-scherm.
7. **Herstart van het hoofdmenu**: Na het weergeven van het bericht van het einde van het spel, wordt het hoofdmenu opnieuw weergegeven zodat de speler een nieuw spel kan starten.

**Hardware Vereisten:**

1. ArduinoUno-bord (Geen Due of ander bord!)
2. Adafruit RGB LCD Shield (16x2 I2C)
3. Drukknoppen met ingebouwde LED's (4 stuks)
4. Weerstanden (bijvoorbeeld 220 ohm) voor de LED's
5. Actieve zoemer (piezo-elektrische luidspreker)
6. Breadboard (optioneel)
7. Jumper wires (aansluitdraden)

**Aansluitingen:**

1. **Adafruit RGB LCD Shield**:
   * Sluit de Adafruit RGB LCD Shield aan op de Arduino Uno zoals aangegeven in de documentatie van Adafruit.
2. **Drukknoppen met LED's**:
   * Sluit elke drukknop aan op een digitale pin van de Arduino Uno (bijvoorbeeld pinnen 6, 7, 8 en 9).
   * Sluit de LED-poten van de drukknoppen aan op de GND van de Arduino via weerstanden (bijvoorbeeld 220 ohm).
   * Verbind de andere poot van elke LED naar de digitale pin die wordt gebruikt voor de LED-bediening.
3. **Piezo-elektrische luidspreker**:
   * Sluit de positieve kant van de zoemer aan op een digitale pin van de Arduino (bijvoorbeeld pin 11).
   * Sluit de negatieve kant van de zoemer aan op de GND van de Arduino.
4. **Optionele Stappen:**

* Als je de drukknoppen en LED's op een breadboard gebruikt, sluit dan eerst de drukknoppen en weerstanden aan op het breadboard en verbind vervolgens de draden met de Arduino.
* Gebruik jumper wires om de verbindingen tussen de componenten te maken volgens de bovenstaande instructies.

Zorg ervoor dat je de componenten voorzichtig aansluit om beschadiging te voorkomen. Controleer ook of de pinnen van de LCD Shield correct zijn uitgelijnd met de pinnen van de Arduino. Als alles correct is aangesloten, kun je de code uploaden naar de Arduino en het spel starten. Veel plezier met spelen!

1. **Adafruit RGB LCD Shield**:
   * In de code wordt de Adafruit\_RGB-LCDShield-bibliotheek gebruikt, wat aangeeft dat de LCD wordt aangestuurd via de I2C-interface.
   * Er is geen expliciete bedrading in de code voor de LCD, omdat deze is verbonden met de I2C-interface van de Arduino via de shield. Dit vereenvoudigt de bedrading aanzienlijk.
2. **Drukknoppen met LED's**:
   * Er zijn vier drukknoppen met LED's gebruikt voor het spel.
   * De drukknoppen worden geactiveerd wanneer ze worden ingedrukt en kunnen ook LED's bevatten die oplichten om feedback te geven.
   * In de functie **initializePins()** worden de pinnen geïnitialiseerd voor de drukknoppen (input) en de LED's (output).
   * De drukknoppen zijn verbonden met de digitale pinnen 6, 7, 8 en 9 van de Arduino, zoals gespecificeerd in de functie **showMainMenu()** waar de knoppen worden gelezen.
   * De LED's van de drukknoppen zijn verbonden met de digitale pinnen en worden aangestuurd om feedback te geven aan de gebruiker in de functie **buttonFeedback()**.
3. **Piezo-elektrische luidspreker**:
   * Een actieve zoemer (piezo-elektrische luidspreker) wordt gebruikt om tonen af te spelen.
   * De zoemer is verbonden met digitale pin 11 van de Arduino, zoals aangegeven in de functie **playNotes()**.
   * Door verschillende frequenties naar de zoemer te sturen, worden verschillende tonen gegenereerd om het spelverloop aan te geven.

De code biedt een duidelijk inzicht in hoe de hardwarecomponenten worden gebruikt en aangesloten. Het benutten van de Adafruit RGB LCD Shield vereenvoudigt de bedrading voor het LCD-scherm, terwijl de drukknoppen en LED's direct op de Arduino worden aangesloten om de interactie met de gebruiker mogelijk te maken. De piezo-elektrische luidspreker wordt gebruikt voor auditieve feedback tijdens het spel.

Bovenkant formulier

**Simon’s Brainfog Game V3.1**

**Afbeelding met tekst, Elektronische engineering, elektronica, stroomkring

Automatisch gegenereerde beschrijving**

**Arduino Connections**

D2 Led 1 OUTPUT Yellow color code

D3 Led 2 OUTPUT Blue color code

D4 Led 3 OUTPUT Green color code

D5 Led 4 OUTPUT Red color code

D6 S1 INPUT Yellow color code

D7 S2 INPUT Blue color code

D8 S3 INPUT Green color code

D9 S4 INPUT Red color code

11 Piezo SPKR OUTPUT Orange color code

A4 - SDA LCD DISPLAY DATA Yellow color code

A5 - SCL LCD DISPLAY CLOCK Blue color code

**LCD Display Connections (I2C)**

**Afbeelding met elektronica, Elektronische engineering, Elektronisch onderdeel, Stroomkringonderdeel

Automatisch gegenereerde beschrijving**

**This example uses an Adafruit 16x2 LCD monochrome display with blue backlight and I2C connected.**

**GAME ANALYSIS**

Initialization: Pins are initialized for input and output. The LCD screen and the speaker are initialized.

1. Main Menu Display: The main menu is displayed on the LCD, and the user can select a difficulty level by pressing the corresponding buttons.
2. Button Feedback: When a difficulty level is selected, a tone is played, the selected push button lights up briefly, and the selected difficulty level is briefly displayed on the LCD screen.
3. Game Start: After selecting a difficulty level, the game begins. A message indicating the start of the game is displayed on the LCD screen.
4. Game Generation: A random sequence of tones is generated based on the selected difficulty level.
5. Game Progression: The Arduino plays the generated sequence of tones, and the player is prompted to repeat the sequence. If the player repeats the sequence correctly, the game continues; otherwise, the game ends.
6. End of Game Approach: If the player loses, they are given the option to play again. If the player wins, their final score is calculated based on the number of completed rounds and any bonus points based on the difficulty level. The final score is displayed on the LCD screen.
7. Main Menu Restart: After displaying the end of game message, the main menu is displayed again so the player can start a new game.

**Hardware Requirements:**

1. Arduino Uno board (No Due or other boards!)
2. Adafruit RGB LCD Shield (16x2 I2C)
3. Push buttons with built-in LEDs (4 pieces)
4. Resistors (e.g., 220 ohms) for the LEDs
5. Active buzzer (piezoelectric speaker)
6. Breadboard (optional)
7. Jumper wires

**Connections:**

1. Adafruit RGB LCD Shield: • Connect the Adafruit RGB LCD Shield to the Arduino Uno as indicated in the Adafruit documentation.
2. Push Buttons with LEDs: • Connect each push button to a digital pin of the Arduino Uno (e.g., pins 6, 7, 8, and 9). • Connect the LED legs of the push buttons to the GND of the Arduino via resistors (e.g., 220 ohms). • Connect the other leg of each LED to the digital pin used for LED control.
3. Piezoelectric Speaker: • Connect the positive side of the buzzer to a digital pin of the Arduino (e.g., pin 11). • Connect the negative side of the buzzer to the GND of the Arduino.
4. Optional Steps: • If using the push buttons and LEDs on a breadboard, first connect the push buttons and resistors to the breadboard and then connect the wires to the Arduino. • Use jumper wires to make connections between the components according to the above instructions. Ensure to connect the components carefully to prevent damage. Also, check if the pins of the LCD Shield are correctly aligned with the pins of the Arduino. Once everything is properly connected, you can upload the code to the Arduino and start the game. Have fun playing!
5. Adafruit RGB LCD Shield: • The code utilizes the Adafruit\_RGBLCDShield library, indicating that the LCD is controlled via the I2C interface. • There is no explicit wiring in the code for the LCD since it's connected to the I2C interface of the Arduino via the shield. This significantly simplifies the wiring.
6. Push Buttons with LEDs: • Four push buttons with LEDs are used for the game. • The push buttons are activated when pressed and may also include LEDs that light up to provide feedback. • In the initializePins() function, the pins are initialized for the push buttons (input) and the LEDs (output). • The push buttons are connected to digital pins 6, 7, 8, and 9 of the Arduino, as specified in the showMainMenu() function where the buttons are read. • The LEDs of the push buttons are connected to the digital pins and are controlled to provide feedback to the user in the buttonFeedback() function.
7. Piezoelectric Speaker: • An active buzzer (piezoelectric speaker) is used to play tones. • The buzzer is connected to digital pin 11 of the Arduino, as indicated in the playNotes() function. • By sending different frequencies to the buzzer, different tones are generated to indicate the game progression. The code provides a clear insight into how the hardware components are used and connected. Utilizing the Adafruit RGB LCD Shield simplifies the wiring for the LCD screen, while the push buttons and LEDs are directly connected to the Arduino to enable user interaction. The piezoelectric speaker is used for auditory feedback during the game.