Grundlage der Entwicklung: Beschreibung der Anforderungen im Lastenheft

**1. Zielbestimmung**

Es soll mit Hilfe des nodeMCU-Microcontrollerboards das Spiel „Simon Says“ (Senso) als mobile Spielkonsole gebaut werden. Hierzu soll sowohl die Hardwareschaltung als auch die Software erstellt werden.

**2. Produkteinsatz**

Das Produkt dient als Proof-of-Concept (kleiner Ausschnitt eines Großprojekts) und soll nicht produktiv eingesetzt werden. Das Produkt soll modular entwickelt werden.

**3. Produktfunktionen**

**Anforderungen**

|  |  |
| --- | --- |
| **Pflichtmodul** |  |
| /LF10/ Das Spiel besteht aus vier großen Feldern in den Farben Rot, Blau, Gelb und Grün, jedem Feld ist eine LED und eine Taste zugeordnet. |  |
| /LF20/ Die Felder leuchten in einem zufällig ausgewählten, aber innerhalb eines Spiels identischen Musters (beginnend mit einem einzigen Feld). Der Spieler muss die Taster der Felder in der vom Muster vorgegebenen Reihenfolge drücken. |  |
| /LF30/ Sofern das Muster durch den Spieler korrekt Eingegeben wurde, verlängert sich das Muster um ein weiteres zufällig ausgewähltes Feld. |  |
| /LF40/ Die Länge der erfolgreich durch den Spieler nachgespielten Reihenfolgen stellt dessen erreichte Punkte dar. |  |
|  |  |
| **Bonusmodul** |  |
| /LF50/ Mit der Länge der Reihe nimmt auch die Geschwindigkeit der Reihen zu. |  |
| /LF60/ Die Ausgangsgeschwindigkeit lässt sich über ein Potentiometer einstellen. |  |
| /LF70/ Der aktuelle Punktestand wird auf dem Dot-Matrix-Display ausgegeben. |  |
| /LF80/ Mehrbenutzerbetrieb: Zu Spielbeginn wird angegeben, wie viele Spieler teilnehmen. Hat der erste Spieler das Muster korrekt eingegeben, ist der zweite Spieler am Zug. Das Display gibt aus, welcher Spieler am Zug ist. Der Punktestand wird für jeden Spieler separat erfasst. |  |
| /LF90/ Levelsystem I: Es besteht die Möglichkeit im Expertenmodus zu spielen: Die Reihenfolgen werden vorwärts vorgespielt und müssen Rückwärts nachgespielt werden. |  |

Legende: O geplant X umgesetzt ✓ getestet

**4. Produktdaten:**

/LD10/ Es muss keine Speicherung von Spielständen über den Neustart hinaus erfolgen (Persistenz).

**5. Qualitätsanforderungen:**

/LQ10/ Fehleingaben mit der Tastatur oder der Maus sollen abgefangen werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Produktqualität | sehr wichtig | eher wichtig | eher unwichtig | nicht wichtig |
| Funktionalität | x |  |  |  |
| Benutzbarkeit |  | x |  |  |
| Zuverlässigkeit |  | x |  |  |
| Effizienz |  |  | x |  |
| Änderbarkeit |  | x |  |  |
| Übertragbarkeit |  | x |  |  |

**6. Ergänzungen:**

|  |  |
| --- | --- |
| **Dokumentation** |  |
| Der aktuelle Quellcode, das ausgefüllte Lastenheft und die Kann-Liste werden nach jedem Termin im ePortfolio Mahara aktualisiert. |  |
| Der Softwareentwurf wird als Programmablaufplan dokumentiert. |  |
| Die Wahl der Variablen- und Methodennamen erfolgt eindeutig beschreibend. Jede Zeile des Programmcodes, die nicht aufgrund von Variablennamen und Methodennamen schlüssig ist, wird kommentiert. |  |
| Das Schaltungslayout wird über ein Fritzing-Technologieschema und über einen händisch zu zeichnenden Schaltplan dokumentiert (alle vorhandenen Bauteile dürfen im Schaltplan als BlackBox zusammengefasst werden). |  |
|  |  |