

PHÔS² – The Interactive Square

Dokumentation

Georgie

Master Medieninformatik – Universität Ulm

Sketching with Hardware II (WS 24/25)

1. Einleitung

Namensherkunft: Die Bezeichnung „Phôs“ (φως) stammt aus dem Griechischen und bedeutet „Licht“. Da das Spielfeld eine Fläche von 1 m × 1 m misst, wurde ein Quadrat-Symbol (²) hinzugefügt. Damit verdeutlicht **PHÔS²**, dass Licht (phôs) im quadratischen Format (1×1 m) interaktiv genutzt wird.

Moderne interaktive Systeme kombinieren physische und digitale Elemente, um immersive Spielerfahrungen zu schaffen. Das Projekt **PHÔS²** ist ein modularer interaktiver Reaktionstisch, der kapazitive Touchsensoren, LED-Ringe und WLAN-Steuerung integriert. **PHÔS²** bietet drei Modi, die im Quellcode klar definiert sind:

1. **Game Mode** – Reaktionsspiel mit Zeitlimit und automatischer Punktevergabe.
2. **Memory Mode** – Farb-Sequenzen merken und wiederholen.
3. **Ambient Mode** – Reaktiver Tischmodus mit Lichtanimationen.

PHÔS² nutzt ein durchdachtes Sensorkonzept zur Berührungserkennung und kombiniert dies mit einer intelligenten LED-Steuerung, die für eine nahtlose visuelle Rückmeldung sorgt. Das Materialdesign sorgt für eine langlebige und robuste Konstruktion, während die Software-Architektur darauf ausgelegt ist, flexibel erweitert zu werden. Die WLAN-Steuerung ermöglicht einen Echtzeit-Moduswechsel über ein Web-Interface und bietet den Spielern damit die Möglichkeit, direkt im Browser zwischen den verschiedenen Spielmodi (Game, Memory, Ambient) zu wechseln.

2. Zielsetzung

Ziel des Projekts ist die Entwicklung eines interaktiven Reaktionsspiels mit:

1. Physisch-digitale Hybrid-Erfahrung durch kapazitive Sensoren & LEDs.
 2. Hochwertige Verarbeitung mit MDF, Plexiglas und eingelassener Elektronik.
 3. WLAN-Steuerung für Modus Verwaltung.
 4. Mehrere Spielmodi (Game/Memory/Ambient) per Software auswählbar.
 5. Interaktives Echtzeit-Feedback (Touch → LED).
-

3. Technische Spezifikationen

3.1 Hardware

Komponente	Details
Mikrocontroller	Arduino Mega 2560 (ATmega2560, 16 MHz)
Touch-Sensoren	16 × kapazitive Touch-Sensoren (F47-A)
LED-Ringe	16 × Ringe (WS2812B), je 35 LEDs, 5 V
LED-Streifen (Strips)	9 × 1m Streifen mit jeweils 60 LEDs (WS2812B)
Stromversorgung	5 V / 70 A-Netzteil + MT3608 Step-Up (5→7 V) für Arduino
WLAN	ESP8266 (via WiFiEsp-Library)

4. Konstruktion & Fertigung

4.1 Materialien

- MDF-Gestell: 4 × Platten (1,0 × 1,0 m, 3 mm), ursprünglich war CNC-Fräsen geplant, jedoch wurde aufgrund fehlender CNC-Kapazitäten mit einem Lochbohrer gearbeitet. Die einzelnen Platten wurden anschließend passgenau zusammengeleimt.
- Plexiglas: 2,5 mm dick.
- Kabel & Shield: Kabel mit Schraubklemmen am Shield fest verschraubt.

4.2 Fertigungsverfahren

- MDF-Zuschnitt: Anstelle einer großen CNC-Fräse (1×1 m) kamen Lochbohrer zum Einsatz, um die Bohrungen für LED-Ringe, Sensoren und Kabeldurchführungen herzustellen. Die einzelnen Platten wurden dann übereinander geleimt, sodass die Elektronik versenkt im MDF liegt.
 - Integration: Sensoren & Ringe werden von unten eingepasst und verklebt.
 - Sicherheit: Kaltgerätebuchse mit Sicherung.
-

5. Funktionsweise

Der Code definiert drei wesentliche Modi, die über das Web-Interface ausgewählt werden:

1. Game Mode (Modus 1 EASY)

- Spieler 1 und Spieler 2 melden sich jeweils bereit (Touch-Sensor am jeweiligen Ring 4).
- Nach dem Countdown leuchten Zufallsringe auf.
- Berührt ein Spieler innerhalb einer gültigen Zeit (< 500 ms) das aufleuchtende Feld, gibt es einen Punkt.
- Cheat-Erkennung: Wenn mehr als 2 Sensoren gleichzeitig berührt werden, wird der Spieler temporär gesperrt.
- Das Spiel endet nach 30 Sekunden.
- Punktestand wird über LED-Strips[1] (Spieler Rot) und LED-Strips[7] (Spieler Blau) angezeigt.

2. Memory Mode (Modus 2 HARD)

- Eine Licht-/Ringsequenz wird zufällig generiert und nacheinander angezeigt.
- Der Spieler muss die Sequenz auf seinen 8 Sensorfeldern wiederholen.
- Bei Fehler oder bei Erreichen von 8 Runden blinkt der Verlierer/Gewinner.
- Spiel wird automatisch zurückgesetzt.

3. Ambient Mode (Modus 3 Chillig)

- Jeder der 8 Sensoren pro Spieler löst bei Berührung durch Hand oder Gegenstand (z.B. Glas) eine zufällige Farbe auf dem entsprechenden LED-Ring aus.
- Zusätzlich laufen zufällige Effekte auf den LED-Streifen (z. B. Zufallsfarben).

5.1 Sensorlogik

- Die Touch-Sensoren sind als *INPUT_PULLUP* definiert, d. h. aktiv LOW.
- Ein Berühren wird über Abfrage *digitalRead(sensorPin) == LOW* erkannt.
- Einfache Entprellung (Debounce) erfolgt mit 20 ms Wartezeit.
- *countTouchedSensors()* erkennt gleichzeitige Berührungen.

5.2 LED-Ansteuerung

- *Adafruit_NeoPixel*-Library für WS2812B.
- Jeder Ring hat 35 LEDs, jeder Streifen 60 LEDs.
- *ShootingEffect*-Struktur für animierte Licht-“Schüsse“ entlang der Streifen.
- *ScoreDisplay* zur Punktanzeige.

5.3 WLAN & Web-Interface

- *WiFiEsp*-Library für ESP8266.
 - Einfache HTTP-Abfrage mit *.../setmode?mode=X* legt Spielmodus fest (1, 2 oder 3).
 - HTML-Formular, um den Modus zu wechseln.
-

6. Ergebnisse & Besonderheiten

1. Game Mode (Speed-Spiel)
 - 30-Sekunden-Zeitfenster.
 - Cheat-Blockierung für 1 Sekunde (*CHEATING_BLOCK_TIME*), wenn mehr als 2 Sensoren berührt.
 - Punkteanzeige in Echtzeit auf LED-Stripe 2 (Rot, Index [1]) bzw. LED-Stripe 8 (Blau, Index [7]).

2. Memory Mode

- Spieler 1 und Spieler 2 haben jeweils unabhängige Sequenzen.
- Bei Fehler blinkt der Gewinner, das Spiel wird zurückgesetzt.
- Maximal 8 Runden.

3. Ambient Mode

- Permanente Abfrage der 8 Sensoren je Spieler.
- Jede Berührung erzeugt eine zufällige Farbe auf dem entsprechenden Ring.
- Zufällige Lichtimpulse auf den Streifen (Standardhelligkeit 50–255).

4. Code-Architektur

- `setup()`: Initialisiert Sensor-Pins, NeoPixel, WLAN.
- `loop()`: Liest Web-Requests, verzweigt je nach `currentMode`.
- `gameModeLoop()` / `memoryGameLoop()` / `ambientTableLoop()`.

5. Evaluation

- Reaktionszeiten < 50 ms (LED-Feedback).
- Hohe Erkennungsrate, Fehlsignale werden durch Debounce & Cheat-Check reduziert.

7. Aufbau & Installation

1. Elektronik

- Arduino Mega + integrierter ESP8266 (WiFiEsp)
- Sensoren an Pins 1–8 (Player1) / 16–21, 31, 32 (Player2)
- LED-Ringe auf Pins 9 & 10, LED-Strips an Pins 22–30.

2. Software

- Bibliotheken: `Adafruit_NeoPixel`, `WiFiEsp`
- Kompilieren in Arduino IDE, Baudrate 115200.

3. Web-Interface

- SSID & Passwort im Code anpassen.
- Nach Verbindungsaufbau via Browser `http://<IPAdresse>` aufrufen.
- Moduswahl über HTML-Formular.

8. Teileliste & Budget

Bauteil	Menge	Stk. Preis	Kosten
Arduino Mega	1	15,29 €	15,29 €
Kapazitive Sensoren (F47-A)	16	4,39 €	70,24 €
Netzteil 5v 70A	1	33,59 €	33,59 €
StepUp Modul		2 €	2 €
16 LED-Ringe (35 LEDs)	16	4,59 €	73,44 €
9 LED-Streifen (60 LEDs)	9	4,99 €	44,91
Plexiglas (1×1 m, 10 mm)	1	50 €	50 €
MDF-Platten	5	8 €	40 €
Konstruktionsholz	3	3,96 €	11,88 €
Kabel und Verbinder	-	10 €	10 €
Summe (ca.)	-		351,35 €

(Die Kosten variieren je nach Zeitraum und Bezugsquelle.)

9. 3D-Modell & Projektzeitplan

3D-Modell

- Ein Autodesk-Modell wurde erstellt, um die benötigten Ausschnitte (für LED-Ringe und Sensoren) auf einer CNC-Fräse oder einem Lasercutter präzise herzustellen.
- Aufgrund fehlender CNC-Kapazitäten im 1000×1000-mm-Format habe ich jedoch auf das Fräsen in einem Stück verzichtet. Stattdessen habe ich 3 × 3 mm MDF-Platten verwendet:
 1. Platte: 100-mm-Löcher für die LED-Ringe (2,8 mm dick, 98 mm Umfang)

2. 2. Platte: 50-mm-Löcher für die Touchsensoren (6 mm dick, 46 mm Umfang)
 3. 3. Platte: Kabeldurchführungen anschließend wurden alle drei Platten zusammengeleimt, sodass die Elektronik sauber im MDF versenkt ist. Dadurch entstand das gewünschte 1×1-m-Spielfeld.
- Theoretisch könnte man den Prozess mit einer einzigen 9-mm-MDF-Platte und passender CNC-Maschine deutlich beschleunigen und mehrere Arbeitsschritte einsparen.

Projektzeitplan

1. Planung & Konzept (3 Tage)
 - Projektdefinition und Anforderungen
 - Skizzen und CAD-Modelle (Autodesk)
 - Materialliste und Bestellung der Komponenten
 2. Aufbau der Struktur (1 Woche)
 - MDF-Gestellbau
 - Installation der Plexiglasscheibe
 - Schutzprüfung
 3. Elektronik & Programmierung (2–4 Wochen)
 - Montage der LED-Ringe & Touch-Sensoren
 - Arduino-Programmierung (Spielmodi, LED-Steuerung)
 - WLAN-Setup und Test
 4. Integration und Tests (1 Woche)
 - Einbau und Verkabelung aller Komponenten
 - Systemtests und Fehlerbehebung
 5. Finalisierung & Dokumentation (1 Woche)
 - Äußere Gestaltung (Branding)
 - Qualitätskontrolle
 - Anfertigung der endgültigen Dokumentation und DIY-Anleitung
-

10. Fazit & Ausblick

Erfüllte Must-have-Kriterien in PHÔS²

✓ Nicht einfach Etwas nachbauen

- **PHÔS²** wurde von Grund auf eigenständig entwickelt und kombiniert verschiedene Technologien zu einem einzigartigen interaktiven Reaktionstisch.
- Die Software-Logik für die drei Spielmodi wurde individuell programmiert, einschließlich Cheat-Erkennung und LED-Effekten.
- Statt fertiger Module wurde ein eigener Aufbau mit MDF-Platten und kapazitiven Sensoren konzipiert.

✓ Microcontroller mit sinnvoller Funktion

- Arduino Mega steuert die Touch-Sensoren und LED-Ringe mit akzeptabler Latenz.

✓ Sinnvolles Fertigungsverfahren

- MDF-Platten wurden mit Lochbohrern präzise bearbeitet und zusammengeleimt.
- Sensoren und Elektronik sind sauber versenkt, was die Haltbarkeit erhöht.

✓ Interaktivität mit menschlichem Input und LED-Feedback

- Kapazitive Touch-Sensoren erkennen Berührungen oder Gegenstände durch Plexiglas.
- Sofortige visuelle Rückmeldung über LED-Ringe und animierte Lichteffekte.

✓ Sicherheitsaspekte berücksichtigt

- Kabel sauber geführt und abgeschirmt.
- Kaltgerätestecker mit Sicherung außen integriert.

Erfüllte Should-have-Kriterien in PHÔS²

✓ Vernetzung über WLAN

- Web-Interface ermöglicht die Moduswahl direkt im Browser.
- Potenzial für eine API-Anbindung zur Ansteuerung über Smart Home oder externe Services.

✓ Über das Prototypenstadium hinaus

- Keine Breadboards, stattdessen fest verlötete Verbindungen von Arduino mit dem Custom Shield und feste Verschraubung der Anschlüsse am Shield.
 - Robuste Konstruktion für langfristige Nutzung.
-

Zukünftige Erweiterungen

1. Soundeffekte oder haptisches Feedback für ein noch intensiveres Spielerlebnis.
 2. Erweiterte API-Anbindung, z. B. für Online-Highscores oder Smart-Home-Integration.
 3. Mehr Sensoren und größere Spielfelder, um das Spiel weiter zu skalieren.
 4. ShootingEffect überarbeiten: Die aktuellen animierten Licht-Schüsse entlang der LED-Streifen könnten für einige Nutzer irritierend sein. Es sollte evaluiert werden, ob eine alternative visuelle Darstellung oder eine deaktivierbare Option sinnvoll wäre.
-

PHÔS² zeigt, wie sich durch den Einsatz von kapazitiven Sensoren, LEDs und einer stabilen Software-Architektur ein innovatives Reaktionsspiel umsetzen lässt. Das System ist modular und kann in Zukunft flexibel erweitert werden.

