**Frankfurt University of Applied Sciences**

Fachbereich Informatik

Projektbericht im Rahmen der Lehrveranstaltung

**Human Computer Interaction (HCI)**

**Color Memory**

Wintersemester 2025/2026

Prof. Dr.-Ing. Matthias Deegener

Autorinnen und Autoren:

Younes Wimmer · John Grosch · Parnia Esfahani

Frankfurt am Main, Januar 2026

# Inhaltsverzeichnis

1. Einleitung

2. Theoretischer Hintergrund

2.1 Wahrnehmung

2.2 Gedächtnis

2.3 Interaktion

2.4 Handlungsmodelle

2.5 Usability und mentale Belastung

3. Konzept und Implementierung

3.1 Projektziel und Forschungsfragen

3.2 Technologie-Stack und Architektur

3.3 Gestaltung der Benutzeroberfläche

3.4 Audio-Feedback und Gamification

4. Analyse nach HCI-Prinzipien

4.1 Heuristische Evaluation

4.2 Barrierefreiheit und Inklusion

4.3 Kognitive Belastung und Lernförderlichkeit

5. Evaluation und Reflexion

5.1 Methodik

5.2 Ergebnisse und Interpretation

5.3 Lessons Learned

6. Fazit

7. Quellenverzeichnis

# 1. Einleitung

Im Rahmen der HCI-Veranstaltung wollten wir ein kleines, aber rundes Projekt entwickeln. Daraus ist Color Memory entstanden – ein Spiel, bei dem wir testen, wie gut man sich Farbwörter merken kann, wenn Schriftfarbe, Wortbedeutung und Hintergrund durcheinandergeraten. Wir haben uns dabei eng an den Inhalten der Vorlesung orientiert und versucht, Theorie in eine praktische Anwendung zu übersetzen [1].

Unser Ziel war kein Hochglanz-Produkt, sondern eine Anwendung, die man gut im Labor oder im Seminar zeigen kann. Wir wollten nachvollziehen, wie Wahrnehmung, Gedächtnis und Interaktion zusammenspielen und wie sehr gutes Feedback die Spielerinnen und Spieler motiviert. Der Bericht begleitet den Prozess von der Idee bis zur Evaluation und greift immer wieder auf das Vorlesungsskript sowie ergänzende Literatur zurück [1][2][3][4].

Wir waren zu dritt an dem Projekt beteiligt und hatten jeweils unterschiedliche Schwerpunkte: Interface-Design, Logik und Evaluation. Die Zusammenarbeit lief iterativ. Wir haben früh Prototypen gebaut, Feedback eingeholt und daraus neue Versionen erstellt. Der Bericht dokumentiert diese Lernkurve Schritt für Schritt.

# 2. Theoretischer Hintergrund

Bevor wir mit dem Design angefangen haben, haben wir die Grundlagen zusammengetragen. Das Vorlesungsskript von Deegener diente als Startpunkt und behandelte Wahrnehmung, Gedächtnis, Handlungsmodelle und Usability verständlich und praxisnah [1]. Zusätzlich haben wir uns auf Baddeleys Arbeiten zum Arbeitsgedächtnis und auf etablierte Usability-Standards gestützt [2][3][4].

Mit diesen Quellen konnten wir konkrete Anforderungen für das Spiel formulieren. Wir wollten zum Beispiel die Gestaltgesetze sichtbar machen, das 7±2-Gedächtnislimit berücksichtigen und gleichzeitig das Interface so simpel halten, dass niemand im Menü hängen bleibt. ISO 9241-110 gab uns klare Leitplanken, welche Dialogprinzipien unbedingt eingehalten werden müssen [4].

In den folgenden Unterkapiteln fassen wir die wichtigsten Punkte aus den Quellen zusammen und zeigen, wie sie unseren Entwurf beeinflusst haben. Für uns war das auch ein Weg, die Vorlesungsinhalte zu wiederholen und direkt auf unser eigenes Projekt anzuwenden.

## 2.1 Wahrnehmung

Das Spiel steht und fällt mit visueller Wahrnehmung. Wir nutzen Farben, Texte und Animationen, damit die Spielenden sofort wissen, in welcher Phase sie sich befinden. Die Gestaltgesetze aus der Vorlesung – vor allem Nähe und Ähnlichkeit – haben wir bewusst eingesetzt. Buttons, die zusammengehören, stehen nah beieinander und sehen ähnlich aus [1].

Wir haben uns außerdem mit Kontrasten beschäftigt, damit die Farbwörter trotz bunter Hintergründe gut lesbar bleiben. Da nicht alle Menschen Farben gleich wahrnehmen, ergänzen wir jede Information textlich. So bleibt die Botschaft erhalten, auch wenn Farben schwer zu unterscheiden sind.

Um das Ganze nicht zu hektisch wirken zu lassen, setzen wir Animationen nur dort ein, wo sie helfen: ein kurzes Aufleuchten der Kachel nach einer Eingabe oder ein sanftes Einblenden des Feedbacks. So bleibt das Interface lebendig, ohne zu stressen.

## 2.2 Gedächtnis

Color Memory fordert vor allem das Kurzzeitgedächtnis. Wir orientieren uns am 7±2-Prinzip, das Baddeley beschreibt [2]. Die Sequenzen werden mit jeder Runde etwas länger, bis irgendwann der Punkt kommt, an dem man sich nicht mehr alles merken kann. Genau da wird es spannend, weil dann Strategien wie Chunking oder laut Mitsprechen ins Spiel kommen.

Das Spiel ist so aufgebaut, dass es beim Lernen hilft. Es gibt klare Instruktionen, die Auswahl wird angezeigt und nach jeder Runde erfährt man sofort, ob man richtig lag. Diese Rückmeldungen sollen das Gedächtnis entlasten, weil man nicht selbst nachzählen muss, wie viele Wörter schon dran waren.

Das Highscore-System funktioniert als kleine Gedächtnisstütze: Wer öfter spielt, sieht schnell, ob sich das Üben lohnt. Außerdem entsteht ein Anreiz, das Spiel mehrfach zu starten und zu schauen, wie weit man kommt.

## 2.3 Interaktion

Unter Interaktion verstehen wir die direkte Kommunikation zwischen Person und System. Wir wollten, dass jede Aktion eine sichtbare Reaktion auslöst. Startet man das Spiel, verschwindet das Menü. Klickt man eine Farbe an, leuchtet sie auf. Diese direkte Manipulation macht das Spielgefühl intuitiv und entspricht Empfehlungen aus der Vorlesung [1].

Die Buttons sind bewusst groß gestaltet und eindeutig beschriftet. Selbst ohne lange Einführung sollte klar sein, welche Funktion sich hinter einem Icon verbirgt. Dazu kommt, dass das Spiel sofort Feedback gibt, wenn etwas nicht passt. Es gibt also keinen Moment, in dem man im Dunkeln tappt.

Auch die Steuerbarkeit war uns wichtig. Man kann das Spiel stoppen, ohne den Fortschritt zu verlieren, und jederzeit ins Menü zurück. Damit erfüllen wir zentrale Punkte der ISO 9241-110 wie Steuerbarkeit und Fehlertoleranz [4].

## 2.4 Handlungsmodelle

Die Handlungsmodelle aus der Vorlesung helfen, den Ablauf einer Runde zu verstehen. Wir orientieren uns vor allem an Norman und den dort beschriebenen Phasen Wahrnehmen, Planen, Ausführen und Bewerten [1]. Unser UI folgt dieser Reihenfolge: erst anschauen, dann merken, danach auswählen, zuletzt Feedback.

Damit der Weg von der Idee zur Aktion kurz ist, haben wir die Buttons so angeordnet, dass keine langen Mauswege nötig sind. Außerdem sind die Kacheln während der Memorierphase deaktiviert – das reduziert Fehlversuche und passt zu den Handlungsschleifen aus der Theorie.

Wir haben gemerkt, dass die Modelle auch beim Testen helfen. Wenn jemand bei der Evaluation hängengeblieben ist, konnten wir prüfen, an welcher Stelle der mentale Plan unterbrochen wurde. Das half uns bei der Optimierung.

## 2.5 Usability und mentale Belastung

Usability war ein Dauerthema. Wir wollten das Spiel so gestalten, dass es effektiv, effizient und angenehm nutzbar ist – genau wie es im Skript und in ISO 9241-110 beschrieben wird [1][4]. Die mentale Belastung sollte fordernd, aber nicht frustrierend sein. Deshalb ist der Timer standardmäßig aus. Wer Tempo mag, kann ihn einschalten, alle anderen spielen entspannt weiter.

Wir haben versucht, unnötige Belastung zu vermeiden. Es gibt keine überflüssigen Pop-ups, dafür klare Texte und gut sichtbare Buttons. Gleichzeitig sorgen motivierende Rückmeldungen dafür, dass man sich ernst genommen fühlt, auch wenn eine Runde nicht klappt.

Die Kombination aus Farben, Texten und Sounds lenkt die Aufmerksamkeit, ohne zu überfordern. So entsteht eine Lernumgebung, die man flexibel an das eigene Tempo anpassen kann.

# 3. Konzept und Implementierung

In diesem Kapitel beschreiben wir, wie wir das Spiel geplant und umgesetzt haben. Wir starten mit den Projektzielen, erklären den Technologie-Stack und gehen danach auf UI und Audio ein. Viele Entscheidungen sind aus dem Team heraus entstanden, andere haben wir aus der Theorie übernommen.

Color Memory war vom ersten Tag an als Desktop-App gedacht. Gleichzeitig wollten wir moderner wirken als klassische Tkinter-Interfaces. Mit Flet haben wir eine Plattform gefunden, die Flutter-Komponenten per Python zugänglich macht. So konnten wir schnell erste Layouts testen und später verfeinern.

## 3.1 Projektziel und Forschungsfragen

Unsere Kernfragen lauteten: Wie schwer darf das Spiel sein, damit es Spaß macht? Welche Rolle spielt Feedback beim Lernen? Und welche HCI-Regeln helfen uns konkret weiter? Daraus entstanden Anforderungen wie klare Instruktionen, sichtbares Feedback und eine Bedienung ohne lange Einarbeitung [1][3][4].

Wir haben das Projekt als Lernplattform für uns selbst verstanden. Jede Person im Team sollte an einem Teil wachsen: Frontend, Logik, Evaluation. Gleichzeitig wollten wir etwas schaffen, das wir im Seminar stolz präsentieren können – also eine Mischung aus wissenschaftlichem Anspruch und spielerischem Experiment.

## 3.2 Technologie-Stack und Architektur

Technisch basiert alles auf Python 3.12. Flet liefert die UI-Komponenten, und im Hintergrund kümmert sich unsere eigene Engine um die Spielstände. Die Dateien sind klar getrennt: color\_memory.py für die Oberfläche, game.py für Logik, audio.py für Sounds, config.py für Farben und Pfade, utils.py für Hilfsfunktionen.

Wir lehnen uns an das MVC-Muster an. Die Engine ist das Model, Flet stellt den View bereit, und die Methoden im App-Controller verbinden beides. Besonders wichtig war uns, dass der Timer und andere asynchrone Abläufe die Oberfläche nicht einfrieren. Deshalb nutzen wir page.run\_task und sorgen dafür, dass UI-Updates immer wieder im Haupt-Thread landen.

## 3.3 Gestaltung der Benutzeroberfläche

Beim Design wollten wir es freundlich und übersichtlich halten. Pastellfarben, runde Ecken und genügend Abstand sorgen für eine ruhige Optik. Auf der Startseite gibt es bewusst nur das Logo und die wichtigsten Buttons. So sieht man sofort, was zu tun ist.

Das Spielfeld ist in Abschnitte aufgeteilt: oben Statistiken, in der Mitte das Wort, darunter die Kacheln. Auf großen Monitoren bleiben rechts und links automatisch Ränder, damit nichts ausfranst. Wir haben darauf geachtet, dass alles auch mit Touch bedienbar wäre – die Buttons sind groß genug und weit genug auseinander.

Für Barrierefreiheit haben wir Kontraste geprüft und dafür gesorgt, dass Informationen nicht nur über Farbe vermittelt werden. Auch der Timer-Schalter ist klar beschriftet, damit niemand überrascht wird [4].

## 3.4 Audio-Feedback und Gamification

Audio ist optional, aber wir finden, dass die kurzen Töne das Feedback abrunden. Erfolg klingt hell, Fehler eher dumpf. Auf dem Mac spielen wir Sounds über afplay, auf anderen Systemen kann playsound einspringen. Wer Ruhe braucht, schaltet die Musik einfach aus.

Der Highscore und die kleine Zusammenfassung am Ende jeder Runde sind unsere Gamification-Elemente. Wir wollten keinen großen Belohnungsmechanismus, aber einen kleinen Anreiz, weiterzuspielen. Viele Testpersonen fanden es außerdem gut, nach einem Fehler direkt die richtige Lösung zu sehen.

# 4. Analyse nach HCI-Prinzipien

Nachdem die erste stabile Version stand, haben wir unser Design mit bekannten HCI-Kriterien abgeglichen. Dazu gehören die heuristischen Regeln von Nielsen und die Dialogprinzipien der ISO 9241-110. So konnten wir checken, wo wir schon gut sind und wo noch Luft nach oben ist [3][4].

## 4.1 Heuristische Evaluation

\*Systemstatus sichtbar:\* Runde, Timer und Feedback werden klar angezeigt. Man weiß jederzeit, in welchem Zustand sich das Spiel befindet [3].

\*Sprache der Zielgruppe:\* Wir nutzen einfache Begriffe wie Start oder Zum Menü. Nichts wirkt technisch oder kompliziert.

\*Kontrolle und Freiheit:\* Man kann jederzeit stoppen oder neu starten, ohne das Programm schließen zu müssen. Fehler führen zu einer erklärenden Meldung statt zu einem Absturz [3].

\*Konsistenz:\* Buttons sehen gleich aus, Farben haben klare Bedeutungen. Das Layout ändert sich nicht zwischen den Ansichten.

\*Fehlervermeidung:\* Während der Merken-Phase sind die Kacheln deaktiviert. So klickt niemand versehentlich zu früh.

\*Erkennen statt Erinnern:\* Ausgewählte Farben werden als Text angezeigt. Die Lösung erscheint nach Fehlern. Damit muss man keine verdeckten Zustände im Kopf behalten.

\*Flexibilität:\* Timer und Musik sind optional. Damit kann jede Person das Spieltempo selbst bestimmen.

\*Ästhetik:\* Wir haben auf ein aufgeräumtes Layout geachtet. Alles, was keine Funktion hat, wurde entfernt.

\*Fehlerhinweise:\* Meldungen nennen die Runde und zeigen die korrekte Reihenfolge. Das hilft beim erneuten Versuch.

\*Hilfe:\* README und Bericht erklären Installation und Bedienung. Im Spiel selbst braucht es keine extra Hilfe, weil alles beschriftet ist.

## 4.2 Barrierefreiheit und Inklusion

Unser Ziel war, dass möglichst viele Menschen das Spiel nutzen können. Deshalb haben wir Kontraste geprüft, redundante Infos eingebaut und große Klickflächen verwendet. Auch ohne Farben lässt sich das Spiel spielen, weil die Wörter ausgeschrieben sind.

Der Timer ist freiwillig, damit niemand unter Stress gesetzt wird. Wer gerne schneller spielt, aktiviert ihn. Wer mehr Zeit zum Merken braucht, lässt ihn aus. So reagieren wir auf unterschiedliche Lern- und Spielstile [4].

Sound ist nett, aber kein Muss. Wer nichts hören kann oder will, verpasst keine Information. Gleichzeitig freuen sich Leute mit eingeschränkter Sicht über zusätzliche akustische Hinweise. Für die Zukunft planen wir Tastaturkürzel und Screenreader-Unterstützung.

## 4.3 Kognitive Belastung und Lernförderlichkeit

Wir haben versucht, das Thema mentale Belastung aus der Vorlesung praktisch anzuwenden [1]. Die Aufgabe selbst (Sequenzen merken) ist die inhaltliche Last. Alles drum herum wollten wir schlank halten, damit keine unnötige Zusatzbelastung entsteht.

Das Feedback nach jeder Runde und die klare Struktur der Phasen sollen beim Lernen helfen. Man weiß immer, was als Nächstes kommt, und kann die eigene Leistung einschätzen.

Testerinnen sagten uns, dass das Spiel herausfordernd, aber fair sei. Das sehen wir als Bestätigung, dass die Balance aus Anspruch und Unterstützung ganz gut klappt.

# 5. Evaluation und Reflexion

Wir haben mehrere Testdurchläufe mit Kommilitoninnen und Kommilitonen durchgeführt. Dabei ging es nicht um große Statistik, sondern um ehrliches Feedback: Wo hakt es? Was macht Spaß? Wo wünscht ihr euch Verbesserungen?

Die Tests fanden in zwei Runden statt. Nach der ersten Version haben wir viel Feedback eingebaut und anschließend erneut getestet. So konnten wir direkt sehen, ob sich unsere Änderungen lohnen.

## 5.1 Methodik

Für die qualitative Evaluation haben wir Think-Aloud-Sessions organisiert. Die Teilnehmenden sollten laut sagen, was sie denken, während sie spielen. Wir haben mitgeschrieben und später ausgewertet.

Zusätzlich haben wir einfache Metriken erfasst: Wie viele Runden schaffen sie? Wie oft wird der Timer genutzt? Wie reagieren sie auf Fehler? Diese Daten waren zwar klein, gaben uns aber gute Hinweise.

## 5.2 Ergebnisse und Interpretation

Im Durchschnitt kamen die Testenden bis Runde 6 oder 7. Das passt gut zu den Erwartungen aus dem 7±2-Prinzip und zeigt, dass der Schwierigkeitsgrad halbwegs stimmt [2].

Viele fanden den freiwilligen Timer gut, weil er das Spiel abwechslungsreich macht. Einige wünschten sich zusätzlich eine Hardcore-Variante mit kürzeren Pausen – das haben wir als Idee für die Zukunft notiert.

Kritik gab es anfangs an zu langen Fade-Animationen. Wir haben sie daraufhin verkürzt. Danach fühlte sich das Spiel flüssiger an.

## 5.3 Lessons Learned

Wir haben gelernt, wie wichtig frühes Testen ist. Einige Probleme wären uns allein nie aufgefallen. Think-Aloud hat uns die Augen geöffnet.

Flet war für uns neu, aber durchaus freundlich. Man muss aufpassen, wie man mit Async-Code umgeht, sonst läuft die UI aus dem Ruder. Nach ein paar Fehlversuchen hatten wir den Dreh raus.

Und ganz wichtig: Theorie bringt nur dann etwas, wenn man sie auf die eigene Situation anwendet. Viele Konzepte aus der Vorlesung wurden für uns erst klar, als wir sie im Spiel umgesetzt haben [1].

# 6. Fazit

Color Memory ist ein kleines Spiel, das unsere HCI-Kenntnisse auf den Punkt bringt. Wir haben theoretisches Wissen praktisch getestet und viele Aha-Momente erlebt. Das Spiel eignet sich gut, um im Seminar zu demonstrieren, wie Wahrnehmung, Feedback und Usability zusammenarbeiten [1][3][4].

Wir sehen noch Potenzial für Erweiterungen: adaptive Schwierigkeitsgrade, mehr Accessibility-Features oder vielleicht sogar eine Web-Version. Aber schon jetzt sind wir zufrieden, wie rund die App läuft.

Für uns als Team war das Projekt eine gute Übung in Organisation, Technik und Kommunikation. Wir haben gemerkt, dass gute UX kein Zufall ist, sondern aus vielen kleinen Entscheidungen besteht.

# 7. Quellenverzeichnis

[1] Deegener, M. (2025). Human Computer Interaction. Frankfurt University of Applied Sciences, Vorlesungsskript.

[2] Baddeley, A. (1992). Working memory. Science, 255(5044), 556-559. https://doi.org/10.1126/science.1736359

[3] Nielsen, J. (1994). Usability Engineering. Morgan Kaufmann.

[4] International Organization for Standardization. (2020). ISO 9241-110:2020 – Ergonomics of human-system interaction — Part 110: Interaction principles.