



SOFTWARE- ENTWICKLUNGSPRAKTIKUM (SEP)

ARID - AUGMENTED REALITY IN DISGUISE

Software-Entwicklungspraktikum (SEP)
Sommersemester 2024

Testprotokolle

Auftraggeber:

Technische Universität Braunschweig
Institut für Anwendungssicherheit (IAS)
Prof. Dr. Martin Johns
Mühlenpfordstraße 23
38106 Braunschweig

Betreuerin: Alexandra Dirksen

Auftragnehmer:

Name	E-Mail-Adresse
Amir Fakhim Hashemi	a.fakhim-hashemi@tu-braunschweig.de
Ibrahim Abdullah	i.abdullah@tu-braunschweig.de
Jadon-Kim Fischer	jadon-kim.fischer@tu-braunschweig.de
Mohamed Ali Mrabti	m.mrabti@tu-braunschweig.de
Tim Küttemeyer	t.kuetemeyer@tu-braunschweig.de
Vyvy Nguyen	Vyvy.nguyen@tu-braunschweig.de

Braunschweig, 3. Juli 2024

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1	Jadon	—
2	Vyvy	—
3	Vyvy	—
4	Amir	—
5	Tim	—
6	Mohamed Ali	—
7	Ibrahim	—

Inhaltsverzeichnis

1 Testdurchführung (2024-04-15)	4
1.1 Testumgebung	4
1.2 Testprotokoll	4
1.3 Zusammenfassung	5
2 Testdurchführung (2024-06-12)	6
2.1 Testumgebung	6
2.2 Testprotokoll	6
2.3 Zusammenfassung	7
3 Testdurchführung (2024-06-12)	8
3.1 Testumgebung	8
3.2 Testprotokoll	8
3.3 Zusammenfassung	9
4 Testdurchführung (2024-06-17)	10
4.1 Testumgebung	10
4.2 Testprotokoll	10
4.3 Zusammenfassung	11
5 Testdurchführung (2024-06-29)	12
5.1 Testumgebung	12
5.2 Testprotokoll	12
5.3 Zusammenfassung	13
6 Testdurchführung (2024-06-30)	14
6.1 Testumgebung	14
6.2 Testprotokoll	14
6.3 Zusammenfassung	15
7 Testdurchführung (2024-06-30)	16
7.1 Testumgebung	16
7.2 Testprotokoll	16
7.3 Zusammenfassung	17

1 Testdurchführung (2024-04-15)

Art des Tests: Integrationstest

Ausgeführte Testfälle: **T200**

Beteiligte Tester: Jadon Fischer, Mohamed Ali Mrabti, Tim Küttemeyer

Abgedeckte Funktionen: **F20**

1.1 Testumgebung

Die Durchführung des Testfalls fand auf einem Windows 11-Laptop mit deutscher Systemumgebung statt. Die Teilnehmer platzierten das Monocle-Gerät auf unterschiedlichen Orten, die sich in ihrer Entfernung von dem auf einem Tisch stehenden Laptop unterschieden. Zusätzlich wurde das Monocle-System von den Teilnehmern aufgesetzt um dessen Verbindung im bewegten Zustand zu testen.

1.2 Testprotokoll

Die folgende Tabelle beschreibt die Ausführung des Testfalls.

Testfall	T200
Tester	Jadon Fischer, Mohamed Ali Mrabti, Tim Küttemeyer
Eingaben	Das Monocle-System befindet sich in unmittelbarer Nähe des Host-Geräts (Laptop) und ist eingeschaltet. Auf dem Host-Gerät wird die IDE Visual Studio Code ausgeführt mit der Monocle-Erweiterung von BrilliantLabs. Innerhalb der IDE wird die Host-Anwendung ausgeführt. Die Bluetooth-Funktion des Host-Geräts ist eingeschaltet.
Soll - Reaktion	Die Standard-Oberfläche verschwindet von dem Display des Monocle-Geräts. An dessen Stelle wird die initiale Nutzeroberfläche der eigenen Monocle-Anwendung mit dem konkreten Text „Waiting...” angezeigt.

Ist – Reaktion	Der Text „Waiting...” ist nach mehreren unmittelbaren Versuchen erfolgreich auf dem Display des Monocle-Geräts erschienen. Siehe hierfür „Unvorhergesehene Ereignisse”.
Ergebnis	Die Ist-Reaktion stimmt mit der Soll-Reaktion überein. Somit konnte die Bluetooth Low Energy-Verbindung hergestellt und aufrecht erhalten werden, sodass die Monocle-Anwendung von dem Host-Gerät an das Monocle-Gerät gesendet werden konnte.
Unvorhergesehene Ereignisse	Für die Teilnehmer hat sich während des Testens herausgestellt, dass die maximale Distanz der Bluetooth Low Energy-Kommunikation von 10 Metern nicht überschritten werden sollte, da dies einen Verbindungsabbruch und somit ein Scheitern des Befehls des „Waiting...”-Texts beziehungsweise einen stockenden Einlesevorgang zur Folge hat.
Nacharbeiten	Die Auftragnehmer nehmen die Tüchtigkeit der Bluetooth-Funktion zur Kenntnis und halten bei zukünftigen Tests die maximale Distanz derselben ein.

1.3 Zusammenfassung

Anders als von den Teilnehmern erwartet findet die erste Verbindung zwischen Host-Gerät und Monocle bereits vor der Ausführung des Programms statt. Ebenfalls findet während der gesamten Ausführung der Monocle-Anwendung ein Austausch zwischen Host und Monocle statt, da letzteres wegen der begrenzten Leistung für Arbeitsschritte wie die Erkennung von QR-Codes ungeeignet ist. Die Auftragnehmer starten die Anwendung in unterschiedlichen Distanzen zwischen Host und Monocle und variieren diese auch bei ausgeführter Anwendung. Die Teilnehmer berken dabei die Maximaldistanz der Bluetooth-Kommunikation von 10 Metern. Sie schätzen das Risiko desselben Problems durch den Kunden von der Häufigkeit als vorstellbar und von der Hinderlichkeit als unwesentlich ein, versuchen jedoch den Kunden auf den Sachverhalt hinzuweisen um den Schaden zu minimieren. Die Einschränkung der Softwarequalität durch den genannten Sachverhalt liegt im Bereich der Gebrauchsqualität, dessen Zufriedenheit dadurch beeinträchtigt werden kann. Die Auftragnehmer erörtern deshalb die Sinnhaftigkeit einer mobilen Anwendung, die als Host-Gerät leichter mit dem Monocle mitgeführt werden kann, um Zufriedenheit und Produktivität des Produkts auf diese Weise zu erhöhen. Nicht getestet wurden Befehle an das Monocle-Gerät, die in der Anwendung keine Verwendung finden.

2 Testdurchführung (2024-06-12)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T306**, **T700**

Beteiligte Tester: Vyvy Nguyen, Mohamed Ali Mrabti

Abgedeckte Funktionen: **F50**

2.1 Testumgebung

Monocle befindet sich in unmittelbarer Nähe des Host und ist eingeschaltet, QR-Code wurde gescanned.

2.2 Testprotokoll

Die folgende Tabelle beschreibt die Ausführung des Testfalls.

Testfall	T306, T700
Tester	Vyvy Nguyen, Mohamed Ali Mrabti
Eingaben	Das Monocle befindet sich in unmittelbarer Nähe des Host-Geräts (Laptop) und ist eingeschaltet. Auf dem Host-Gerät wird die IDE Visual Studio Code ausgeführt mit der Monocle-Erweiterung von BrilliantLabs. Monocle und Host sind über BLE mit einander verbunden. Innerhalb der IDE wird die Host-Anwendung ausgeführt. Der QR-Code wurde abfotografiert.
Soll - Reaktion	Die angezeigte verschlüsselte Nachricht entspricht der erwarteten Nachricht, diese sollte vollständig auf dem Monocle und Host dargestellt werden. Wenn der QR-Code nicht gelesen wird soll 'No Data Found' angezeigt werden.
Ist – Reaktion	Die QR-Codes die entschlüsselt werden konnten, zeigten sowohl die Nachricht auf dem Monocle als auch auf dem Host an.
Ergebnis	Der Test ist relativ erfolgreich gewesen. Die Nachrichten sind auf dem Display zusehen.

Unvorhergesehene Ereignisse	Keine da sowohl gelesene als auch fehlgeschlagene Nachrichten angezeigt werden.
Nacharbeiten	Für die bloße Anzeige des Displays wird keine Nachbearbeitung benötigt.

2.3 Zusammenfassung

Testfall 306, 700 überprüft ob die entschlüsselten Nachrichten richtig auf dem Display angezeigt werden. Es wurden auch einfache unverschlüsselte QR-Codes mit dem Monocle abfotografiert und geschaut, ob diese auch angezeigt wurden. Diese Nachrichten wurden auch auf dem Display angezeigt. Solange der QR-Code richtig erkannt und entschlüsselt wurde, konnten die Nachrichten sowohl auf dem Display als auch auf dem Host dargestellt werden.

3 Testdurchführung (2024-06-12)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T300**

Beteiligte Tester: Vyvy Nguyen, Tim Kütemeyer, Mohamed Ali Mrabti

Abgedeckte Funktionen: **F10**

3.1 Testumgebung

Das geladene Monocle liegt im Gehäuse.

3.2 Testprotokoll

Die folgende Tabelle beschreibt die Ausführung des Testfalls.

Testfall	T300
Tester	Vyvy Nguyen, Tim Kütemeyer, Mohamed Ali Mrabti
Eingaben	Die Tester nehmen das Monocle aus dem Gehäuse raus und legen ihn zum Ausschalten wieder hinein.
Soll - Reaktion	Beim Herausnehmen aus dem Gehäuse geht das Monocle an, das Display ist aktiv und beim Hineinlegen geht dieser wieder aus.
Ist – Reaktion	Beim Herausnehmen aus dem Gehäuse geht das Monocle an, das Display ist aktiv und beim Hineinlegen geht dieser wieder aus.
Ergebnis	Das Testen von An- und Ausschalten war erfolgreich.
Unvorhergesehene Ereignisse	Keine
Nacharbeiten	Keine

3.3 Zusammenfassung

Der Testfall T300, T100 überprüft, ob das Monocle ohne Probleme an und aus geht. Der Test lief ohne unvorhergesehene Ereignisse, es ist jedoch wichtig zu erwähnen, dass die Vorbedingungen erfüllt sein müssen und der Akku des Monocles muss geladen sein.

4 Testdurchführung (2024-06-17)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T500**

Beteiligte Tester: Amir Fakhim Hashemi

Abgedeckte Funktionen: **F50**

4.1 Testumgebung

Der Host Computer muss eingeschaltet sein, mit mindestens Python 3.11 installiert. Ein Texteditor in dem der Code zur Entschlüsselung muss geöffnet sein, sowie das Geheimnis und der Schlüssel müssen in dem Code korrekt eingefügt sein.

4.2 Testprotokoll

Die folgende Tabelle beschreibt die Ausführung des Testfalls.

Testfall	T500
Tester	Amir Fakhim Hashemi
Eingaben	Der Tester gibt das Geheimnis und den Schlüssel in den Code ein .
Soll - Reaktion	Der Code verwendet den eingegebenen Schlüssel um das Geheimnis zu entschlüsseln .
Ist – Reaktion	Der wurde nicht richtig entschlüsselt und es kam eine unleserliche Nachricht als Ausgabe
Ergebnis	Der Test ist nicht erfolgreich. Die ausgegebene Nachricht aus dem Entschlüsselungscode entspricht nicht der Nachricht, die vorher Verschlüsselt wurde.
Unvorhergesehene Ereignisse	Entschlüsselung funktioniert nicht korrekt.

Nacharbeiten	Das Problem führt auf die Reihenfolge zurück, da die Nachricht vor seiner Verschlüsselung komprimiert wurde. In dem Testfall wurde erst dekomprimiert und dann entschlüsselt was zu einer falsch entschlüsselten Nachricht führte.
---------------------	--

4.3 Zusammenfassung

Der Testfall T500 zielte drauf ab, die Funktionalität des Codes der zuständig für die Entschlüsselung der Nachricht ist zu überprüfen. Die Tester stellten fest, dass bei wiederholtem testen der Entschlüsselung, mehrfach nicht die zuvor Verschlüsselte Nachricht ausgegeben wurde.

Das nicht funktionieren des Tests wurde als äußerst kritisch angesehen, da die fehlerhafte Funktion der Entschlüsselung sensibler Daten zu einem Verlust der Vertraulichkeit, Integrität und Verfügbarkeit könnte.

Damit die Funktionalität der Entschlüsselung korrekt funktioniert, muss beim entschlüsseln korrekt auf die Reihenfolge geachtet werden. Bevor der Code verschlüsselt wird, wird er noch vorher komprimiert, was den Vorteil hat weniger komplexe QR-Codes zu erstellen, welche dann weniger erkenntlich in den Bildern eingefügt werden können. Bei der Entschlüsselung muss man sich an die korrekte Reihenfolge wie beim Verschlüsseln halten, damit am ende das Geheimnis korrekt entschlüsselt wird und der Nutzer die korrekte Nachricht sieht.

5 Testdurchführung (2024-06-29)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T800**

Beteiligte Tester: Vyvy Nguyen, Tim Küttemeyer

Abgedeckte Funktionen: **F70**

5.1 Testumgebung

Das Monocle-Gerät ist eingeschaltet, die Kamera ist aktiviert, und ein Test QR-Code ist positioniert.

5.2 Testprotokoll

Die folgende Tabelle beschreibt die Ausführung des Testfalls.

Testfall	T800
Tester	Vyvy Nguyen, Tim Küttemeyer
Eingaben	Der Tester berührt Touchpad A oder B um die Kamera auszulösen.
Soll - Reaktion	ie Kamera nimmt ein klares und präzises Foto des QR-Codes auf. Die Bildqualität ist ausreichend, um den QR-Code mit einem QR-Code Lesegerät zu lesen.
Ist – Reaktion	Die Kamera nahm ein Foto auf, jedoch waren einige Bilder unter bestimmten Lichtbedingungen unscharf. Die QR-Codes konnten in diesen Fällen nicht gelesen werden.
Ergebnis	Der Test ist teilweise erfolgreich. Die Kamera funktionierte grundsätzlich, jedoch gab es Qualitätseinbußen bei ungünstigen Lichtverhältnissen. Die meisten QR-Codes konnten gelesen werden, jedoch nicht alle unter allen Bedingungen.
Unvorhergesehene Ereignisse	Unter bestimmten Lichtverhältnissen war die Kamera nicht in der Lage, klare Fotos aufzunehmen.

Nacharbeiten	Verbesserungen in der Bildverarbeitungssoftware sind notwendig, um auch unschärfere Bilder zu erkennen. Die Kamera an sich ist durch die Hardware limitiert.
---------------------	--

5.3 Zusammenfassung

Der Testfall T800 zielte darauf ab, die Funktionalität der Kamera des Monocle-Geräts zu überprüfen, insbesondere ihre Fähigkeit, klare und präzise Fotos von QR-Codes aufzunehmen. Während die Kamera in den meisten Tests zufriedenstellend funktionierte, zeigte sie Schwächen unter gewissen Lichtverhältnissen.

Die Tester stellten fest, dass die Kamera in normaler Umgebung gut funktioniert, jedoch bei heller oder dunkler Beleuchtung Probleme hatte, scharfe Bilder aufzunehmen. Diese Einschränkungen wurden als kritisch für die Funktionalität des Geräts identifiziert, da sie die Zuverlässigkeit der QR-Code-Erkennung beeinträchtigen können.

Zur Verbesserung der Kamera wird empfohlen, die QR Code Erkennungssoftware zu optimieren und da die Kameraeinstellungen sich nicht anpassen lassen. Dies sollte die Gesamtleistung und Zuverlässigkeit des Monocle-Geräts verbessern.

6 Testdurchführung (2024-06-30)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T400**, **T600**

Beteiligte Tester: Mohamed Ali Mrabti, Jadon Fischer

Abgedeckte Funktionen: **F40**

6.1 Testumgebung

Die Durchführung des Testfalls fand auf einem Windows 11 Laptop mit deutscher Systemumgebung, und mit Hilfe von dem IDE Visual Studio Code statt. Das Monocle-Gerät wurde verwendet, um verschiedene QR-Codes aus einer bestimmten Entfernungen zu scannen.

6.2 Testprotokoll

Die folgende Tabelle beschreibt die Ausführung des Testfalls.

Testfall	<i>T400 T600</i>
Tester	<i>Mohamed Ali Mrabti, Jadon Fischer</i>
Eingaben	<i>Monocle eingeschaltet, QR-Code wird aus einer bestimmten Entfernung gescannt.</i>
Soll - Reaktion	<i>Monocle erkennt den normalen oder verschlüsselten QR-Code und zeigt die Nachricht im Terminal sowie auch im Display.</i>
Ist – Reaktion	<i>QR-Code erfolgreich erkannt und korrekte (verschlüsselte) Nachricht angezeigt.</i>
Ergebnis	<i>Während des Tests stellten die Teilnehmer fest, dass die Raumhelligkeit nicht zu hell sein darf und der QR-Code gut auf die Kamera des Monocle ausgerichtet sein muss.</i>
Unvorhergesehene Ereignisse	<i>Der QR-Code sollte eine gute Bildqualität haben und die Raumhelligkeit darf nicht zu hell sein.</i>

Nacharbeiten	Die Auftragnehmer erkennen an, dass der Raum nicht zu hell sein sollte, und achten bei weiteren Tests darauf, dass der QR-Code richtig ausgerichtet ist.
---------------------	--

6.3 Zusammenfassung

Nach erfolgreicher Verbindung des Monocle mit unserem Endgerät beginnt das Monocle nach Berühren eines der Touchpads mit dem Scannen des QR-Codes. Die Auftragnehmer fanden heraus, dass man den (verschlüsselter)QR-Code aus einer bestimmten Entfernung und in einem nicht zu hellen Raum scannen kann, sodass die Kamera des Monocle ein klares Bild des QR-Codes aufnehmen kann und unsere Software die Nachricht korrekt dekodieren kann.

7 Testdurchführung (2024-06-30)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T300**

Beteiligte Tester: Ibrahim Abdullah, Mohamed Ali Mrabti

Abgedeckte Funktionen: **F30**

7.1 Testumgebung

Das Monocle-Gerät ist eingeschaltet und mit dem Host verbunden.

7.2 Testprotokoll

Die folgende Tabelle beschreibt die Ausführung des Testfalls.

Testfall	T300
Tester	Ibrahim Abdullah, Mohamed Ali Mrabti
Eingaben	Der Tester berührt das Touchpad A, B oder beide zusammen.
Soll - Reaktion	Beim Berühren von Touchpad A, B oder beiden zusammen erscheint auf dem Display "Capturing..."
Ist – Reaktion	Beim Berühren von Touchpad A, B oder beiden zusammen erschien jedes Mal auf dem Monocle Display 'Capturing', was zeigt, dass die Touchpad-Berührung funktioniert.
Ergebnis	Der Test der Funktionalität der Touchpads A, B war erfolgreich.
Unvorhergesehene Ereignisse	Keine
Nacharbeiten	Keine

7.3 Zusammenfassung

Der Testfall T300 überprüft die fehlerfreie Funktion des Touchpads unter verschiedenen Benutzereingaben, wie das Berühren von Touchpad A, B oder beiden zusammen. Bei jeder Berührung erschien auf dem Moncole Display die Anzeige “Capturing”, was die ordnungsgemäße Funktion der Touchpads bestätigt. Es traten keine unerwarteten Ereignisse auf.