# Lern- und Arbeitsauftrag Codes, NoSQL und Backend

|  |  |
| --- | --- |
| **Titel:** | **Modul 114, 295, 165 - Einzel/Gruppenarbeit** |
| **Modul:** | IMS-Lernatelier 2 Informatiker/in EFZ |
| **Autor / Version:** | Michael Schneider |
| **Sozialform** | Gruppenarbeit mit 1 bis 3 Personen |
| **Hilfsmittel:** | * Alle |

## Ausgangslage

In diesem Auftrag sollen folgende Module vertieft werden:

Modul 114: Codierungs-, Kompressions- und Verschlüsselungsverfahren einsetzen

Modul 295: Backend für Applikationen realisieren

Modul 165: NoSQL-Datenbanken einsetzen

Suchen Sie sich ein Projekt aus, das zu Ihrem Modulablaufplan passt oder wählen Sie ein eigenes Projekt zur Repetition des Stoffes oder zum Kennenlernen von neuen Technologien aus. Besprechen Sie eigene Projekte mit der Lehrperson.

Hinweis: Es ist nicht erlaubt, Aufgaben für das Qualifikationsverfahren (LB, LBV, …) während des Lernateliers zu realisieren. Das Lernatelier ist auch keine Aufgabenstunde.

Hinweis: Dimensionieren Sie Ihr Projekt so, dass Sie in der gegebenen zeit fertig werden, aber auch genügend Arbeit vorhanden ist, um die Zeit zu füllen.

## Ziele

Am Schluss dieses Lern- und Arbeitsauftrages sollen Sie ein fertiges Produkt mit Projektdokumentation erstellt haben.

## Aufgaben

### 1) Informieren

a) Schliessen Sie sich in Gruppen zusammen, die mindestens aus einer, maximal aus drei Personen bestehen.

b) Wählen Sie in der Gruppe ein Projekt aus. Beachten Sie den Schwierigkeitsgrad, Ihren Wissensstand und die zur Verfügung stehende Zeit bei der Auswahl. Sprechen Sie eigene Ideen mit der Lehrperson ab.

c) Machen Sie eine Anforderungsanalyse. Werden Sie sich in der Gruppe einig, was Sie genau realisieren möchten. Das Projekt sollte durch die Anforderungen so präzise wie möglich beschrieben sein. Legen Sie die minimalen Anforderungen und «nice-to-have»-Anforderungen fest, die bei genügend Zeit realisiert werden können.

d) Legen Sie fest, wo und wie Sie in der Gruppe Arbeitsergebnisse speichern und austauschen möchten.

<https://github.com/Kappa-X/LuckyFlaette>

e) Erarbeiten Sie sich noch fehlende Grundlagen für das Projekt.

|  |  |  |
| --- | --- | --- |
| Nr. | Anforderung | Minimal oder nice to have |
| 1 | Der User kann mit einem Button die Slot-Maschine benutzen | minimal |
| 2 | Der User kann von der Startseite aus zur Spiel Seite | minimal |
| 3 | Der User kann von der Spielseite zur Startseite | minimal |
| 4 | Die Seite ist übersichtlich gestaltet. | Nice to have |
| 5 | Der Gewinn/Verlust ist immer anders, je nach dem, was man dreht. | minimal |
| 6 | Es gibt eine Infoanzeige wo man sieht was wie viele Punkte gibt | Nice to have |
| 7 | Auf der Startseite sieht man verschieden Infos über das Spiel und uns. | Nice to have |
| 8 | Das Login interagiert mit einer API | Minimal |
| 9 | Mit der Stop-Taste kann man das Drehen vom Slot beenden | Minimal |
| 10 | Man kann die Geldmenge eingeben, die man einsetzen will. | Minimal |
| 11 | Es gibt ein Button, wo man die Geldmenge bestätigt. | Minimal |
| 12 | Mit WebPlayer das Unity Spiel auf unsere Seite übernehmen | Minimal |
| 13 | Das Nav Menü öffnet sich mit einem Hamburgermenü | Nice to have |
| 14 | Das Registrieren agiert mit der API | Minimal |
| 15 | Das Passwortändern agiert mit der API | Minimal |
| 16 | Dem aktuellen Benutzer werden seine Geldmenge und Username auf der Hauptseite angezeigt | Nice to have |

### 2) Planen

a) Teilen Sie das Projekt in klare, einzelne Aufgaben auf («Arbeitspakete»).

b) Verteilen Sie diese Arbeiten an die Gruppenmitglieder und setzen Sie für jede Arbeit ein Abschlussdatum fest, dass mit dem Plan Ihrer Klasse übereinstimmt.

c) Erstellen Sie eine ToDo-Liste mit verantwortlicher Person, Arbeitspaket und Abschlussdatum für das ganze Projekt.

|  |  |  |
| --- | --- | --- |
| Arbeitspaket | Datum | Gruppenmitglied |
| Die Startseite erstellen mit HTML und CSS | 21.12.2022 | Florian Portmann/Valentino Belluscio |
| Die Gameseite erstellen mit HTML und CSS | 11.01.2023 | Florian Portmann/Valentino Belluscio |
| Dokumentation erstellen | 25.01.2023 | Valentino Belluscio/Florian Portmann |
| Mit WebPlayer das Unity Spiel auf die Spielseite hochladen | 18.01.2023 | Florian Portmann/Valentino Belluscio |
| Design erstellen für SlotMaschiene auf Unity | 21.12.2022 | Finn Neiger |
| SlotMaschine mit Unity erstellen (Grundaufbau) | 11.01.2023 | Finn Neiger |
| Multiplikator für Slot erstellen mit Unity | 18.01.2023 | Finn Neiger |
| Geld-Setz-Funktion erstellen mit Unity | 25.01.2023 | Finn Neiger |
| API mit Login, Registrieren und Passwort ändern erstellt | 21.12.2022 | Vin Appenzeller |
| Login-Seite mit HTML und CSS | 11.01.2023 | Vin Appenzeller |
| Geldmenge sichtbar auf der Seite mit API | 18.01.2023 | Vin Appenzeller |
| Clientseitiges JavaScript auf Login-Seite und Hauptseite | 25.01.2023 | Vin Appenzeller |

### 3) Entscheiden

Fällen und dokumentieren Sie wichtige Entscheidungen.

Wir haben uns dazu entschieden auf VSC mit Javascript, CSS und HTML zu arbeiten. Auf Unity arbeiten wir mit C#. Währen des Projektes haben wir uns entschieden das Spiel auf itch.io hochzuladen, da WebPlayer veraltet ist und nicht mehr unterstützt wird. Wir hatten keine Zeit mehr, um das Spiel mit WebGL selbst zu hosten.

### 4) Realisieren

Erstellen Sie das Programm und die notwendigen Dokumente. **Erstellen Sie das Programm selbstständig und verwenden Sie nicht einfach das Resultat eines Lernvideos oder fertigen Code.**

Das fertige Produkt ist auf Github. Oben ist das Repository verlinkt.

### 5) Kontrolle

a) Erstellen Sie Testfälle für das Programm.

b) Testen Sie Ihr Programm und schreiben Sie ein kurzes Testprotokoll mit Testumgebung.

|  |  |  |  |
| --- | --- | --- | --- |
| Anforderungs-Nr. | Vorraussetzung | Eingabe | Erwartete Ausgabe |
| 1 | Spielseite ist geladen | Start-Button | Die Slot-Maschine dreht sich |
| 2 | Startseite ist geladen | Button mit Spiel-Verlinkung | Seiten Wechsel von Startseite zur Gameseite |
| 3 | Spielseite ist geladen | Button mit Startseite-Verlinkung | Seiten Wechsel von Gameseite zur Startseite |
| 5 | Slot mehrmals schon gedreht | Button mehrmals | Immer ein neuer Gewinn |
| 6 | Man ist auf der Startseite | Keine Eingabe | Man sieht die Infoanzeige |
| 7 | Man ist auf der Startseite bei About us | Im Nav auf About Us klicken | Man sieht die About Us Section |
| 8.1 | Man ist auf der Login-Seite | Man gibt die richtigen Daten ein und klickt auf den Button | Man wird eingeloggt. |
| 8.2 | Man ist auf der Login-Seite | Man gibt die falschen Daten ein und klickt auf den Button | Man wird nicht eingeloggt |
| 9 | Spielseite ist geladen | Stop-Button | Das Drehen hört auf und der Gewinn/Verlust wird berechnet und angezeigt |
| 10 | Spielseite ist geladen | Menge eingeben | Menge steht im Textfeld |
| 11.1 | Menge ist eingegeben | Button, um Menge hinzuzufügen | Menge wird eingesetzt |
| 11.2 | Keine Menge ist eingegeben | Button, um Menge hinzuzufügen | Error-Meldung |
| 13 | Startseite ist geladen | Hamburgerbutton | Nav wird angezeigt |
| 14 | Login-Seite ist geladen | Daten sind vorhanden und man drückt auf Registrieren | Man ist registriert |
| 15 | Login-Seite ist geladen | Man hat die neuen Daten reingeschrieben und auf den Button gedrückt | Passwort wird verändert |
| 16 | Startseite ist geladen und man ist eingeloggt | Keine Eingabe | Oben sieht man die Geldmenge |

**Testprotokoll:**

|  |  |  |
| --- | --- | --- |
| Testfall Nr. | Durchgeführt von | OK/NOK |
| 1 | Valentino Belluscio | OK |
| 2 | Valentino Belluscio | OK |
| 3 | Valentino Belluscio | OK |
| 5 | Valentino Belluscio | OK |
| 6 | Valentino Belluscio | NOK (haben wir nicht als notwendig empfunden) |
| 7 | Valentino Belluscio | OK |
| 8.1 | Valentino Belluscio | OK |
| 8.2 | Valentino Belluscio | OK |
| 9 | Valentino Belluscio | OK |
| 10 | Valentino Belluscio | OK |
| 11.1 | Valentino Belluscio | OK |
| 11.2 | Valentino Belluscio | OK |
| 13 | Valentino Belluscio | OK |
| 14 | Valentino Belluscio | OK |
| 15 | Valentino Belluscio | OK |
| 16 | Valentino Belluscio | NOK (Das Spiel ist nun auf itch.io und deswegen stimmt die Geldmenge nicht mit der von itch.io überein.) |

**Testbericht:**

**Das Testen verlief gut, denn wir hatten nicht viele Schwierigkeiten. Aber wir haben keine Spielseite selber sondern auf itch.io. Die Spielseite ist trotzdem im Repository drinnen aber, da das WebPlayer Plugin nicht mehr funktioniert geht dies nicht.**

### 6) Auswertung

Überlegen Sie sich, was gut gelaufen ist und was eher nicht. Sie können diese Informationen in Ihrem Portfolioeintrag verwenden.

**Fazit:**

Gut gelaufen: Die Zusammenarbeit in der Gruppe verlief gut und es herrschte eine gute Harmonie zwischen den Gruppenmitgliedern. Das Pair-Programming zwischen Florian und Valentino hat sich auch als gut erwiesen. Wir konnten uns gegenseitig gut unterstützen und lernten neues voneinander.

Schlecht gelaufen: Das Unity Spiel konnten wir nicht via WebPlayer auf unserer Seite hochladen aber wir haben eine andere Lösung gefunden. Denn wir haben es dann mit WebGL auf itch.io hochgeladen. Wir haben keine Zeit mehr gehabt, um das Spiel selbst zu hosten mit WebGL.

### 7) Portfolioeintrag

Schreiben Sie den Portfolioeintrag und reichen Sie ihn ein. Gehen Sie im Portfolioeintrag auf Ihre Ziele und deren Erreichung sowie die erstellte Dokumentation ein. Beschreiben Sie im Portfolioeintrag auch Ihren Lernweg, die Probleme, Erkenntnisse und Erfolge.

## Projektvorschläge

### Modul 114 Codierungs-, Kompressions- und Verschlüsselungsverfahren einsetzen

**Verschlüsselungsapp**

Erstellen Sie eine Applikation, mit der verschiedene Verschlüsselungsverfahren ausgewählt und angewendet werden können, so dass Sie untereinander verschlüsselte Nachrichten und eventuell Bilder austauschen können.

**Steganographie**

Die Steganographie befasst sich damit, Nachrichten zu verstecken, beispielsweise in Bildern.

Erstellen Sie ein Programm mit dem Sie Nachrichten verstecken und wieder auslesen können.

Website: <https://www.loginworks.com/blogs/use-steganography-c/>

Ältere Tutorialreihe: <https://youtu.be/97Vz6wyHeeM>

**Barcode-Erstellung**

Erstellen Sie eine App, mit der Sie Informationen in Barcodes codieren und diese dann mit dem Handy scannen können. Sie können auswählen, ob Sie Bibliotheken verwenden oder die Funktionen selber codieren möchten.

**Zahlensysteme**

Erstellen Sie eine Applikation, mit der Sie benutzerfreundlich Zahlen von einem Zahlensystem in ein anderes umrechnen können.

**Binärcodes**

Erstellen Sie eine Applikation, mit der Sie benutzerfreundlich verschiedene Binärcodes und Zahlen (positiv und negativ, mit Kommas und ohne) ineinander umrechnen können.

**Komprimierung**

Erstellen Sie ein Programm, mit dem Sie eine Datei komprimieren und dekomprimieren können.

### Modul 295: Backend für Applikationen realisieren

Erstellen Sie jeweils ein API für folgende Aufgaben mit einem minimalen Frontend.

Ohne persistente Datenspeicherung (DB):

**Quadratische Gleichungslöser:** Sie haben eine Gleichung wie 3x2 + 12x - 20 = 0. Wenn Sie Ihrem API 3, 12 und -20 übergeben, gibt es die Lösungen zurück.

**Zeitberechner:** Wenn Sie Ihrem API ein Datum übergeben, werden die Sekunden/Minuten/Stunden/Tage zurückgegeben, die zwischen dem aktuellen Zeitpunkt und dem Datum liegen.

**Personenersteller:** Erstellen Sie ein API, das zufällige Personennamen kreiert und zurückgibt.

**Spieleapi:** Erstellen Sie ein API, das

* Schere Stein Papier spielen kann.
* Dem Sie Ihren Einsatz übergeben können und das Roulette spielt.

**Mit persistenter Datenspeicherung (DB):**

**Quiz:** Erstellen Sie eine API für ein Quiz, in dem Sie Fragen und Antworten zu einem bestimmten Themengebiet speichern und sich abfragen lassen können.

**Noten:** Erstellen Sie eine API in dem Sie Ihre Noten Speichern und das Ihnen die Durchschnitzte pro Fach zurückgibt.

**Prüfungen:** Erstellen Sie eine API in dem Sie Daten und Themen Ihrer Prüfungen übergeben können und das Ihnen die verbleibenden Tage zurückgibt.

**Nachrichten:** Erstellen Sie eine API in dem mehrere Personen Nachrichten und Dateien austauschen können. Verschlüsseln Sie die Nachrichten.

### Modul 165: NoSQL-Datenbanken einsetzen

Verwenden Sie für folgende Ideen eine NoSQL Datenbank (MongoDB, Cassandra, …)

**Lernatelierhelper**

Erstellen Sie ein Programm, das Sie nach den wichtigsten Dingen Ihres Projektes fragt und Ihnen ein Gerüst für den perfekten Portfolioeintrag ausgibt.

Integrieren Sie fakultativ eine Rechtschreibekorrektur.

**(Fussball-) Statistikapp**

Suchen Sie online nach Statistiken (beispielsweise über Fussball und Fussballer). Speichern Sie diese in einer NoSQL-Datenbank und erstellen Sie ein Abffrageprogramm.

**Little Google**

Rufen Sie in Ihrem Programm Webseiten ab, folgen Sie den Links und speichern Sie diese in einer NoSQL-Datenbank.

Erstellen Sie eine Applikation, mit der Sie in diesen Webseiten suchen können.

Ebenfalls wäre es möglich, nicht funktionierende Links zu erkennen und auszugeben.

Erweiterung: Erlauben Sie das lokale Browsen dieser Webseiten.

## Gütekriterien

Der Lern- und Arbeitsauftrag ist erfüllt, wenn …

* Wenn Sie das Projekt nach IPERKA abgearbeitet haben.
* Wenn Sie eine lauffähige Applikation produziert haben.

## Zusätzliche Angaben zum Auftrag

Keine.

## Mögliche Erweiterungsaufträge

Keine.