

# Lastenheft - Webanwendung auf Basis von brython und turtle graphics

## Projektdaten

Projektbezeichnung	Webanwendung auf Basis von brython und turtle graphics
Product Owner	Marcus Rosengart
Szenariomitarbeiter	Tom Bohne, Niklas Hagengers, Noah Saibel
Erstellt am	08.10.2023
Letzte Änderung am	09.10.2023
Status	vom Kunden bestätigt
Aktuelle Version	1.3

Tabelle 1: Projektdaten

## Änderungsprotokoll

Version	Datum	Geänderte Kapitel	Art der Änderung	Editor	Status
1.0	08.10.2022	Alle	Erstellung	Tom Bohne, Noah Saibel	-
1.1	09.10.2022	4, 5.1	Anpassung	Noah Saibel	vom Kunden bestätigt
1.2	02.11.2022	5.1	weitere Soll-Ziele	Noah Saibel	vom Kunden bestätigt
1.3	07.11.2022	6	Shell-Scripts zu Lieferumfang	Noah Saibel	vom Kunden bestätigt

Tabelle 2: Änderungsprotokoll

## Inhaltsverzeichnis

<b>1 Ziel des Dokuments</b>	<b>2</b>
<b>2 Ausgangssituation, Zweck und Zielsetzung des Projekts</b>	<b>2</b>
<b>3 Konzept</b>	<b>2</b>
<b>4 Entwurf (grafisch)</b>	<b>2</b>
<b>5 Anforderungen</b>	<b>2</b>
5.1 Funktionale Anforderungen . . . . .	2
5.2 Nichtfunktionale Anforderungen . . . . .	3

## 1 Ziel des Dokuments

- Anforderungen festhalten
  - funktionale Anforderungen
  - nichtfunktionale Anforderung
    - \* Rahmenbedingungen
    - \* Qualitätsanforderungen
- Muss-/Soll-Ziele festhalten

## 2 Ausgangssituation, Zweck und Zielsetzung des Projekts

- Näher bringen an die Programmierung für Kinder im Alter von 10 bis 16 Jahre
- Begrenzte Zeit zur Vermittlung der Programmierkonzepte
  - 1,5 Stunden (während Tag der offenen Tür, Zukunftstag)

## 3 Konzept

- Webanwendung mit einer, für die genannte Zielgruppe, benutzerfreundliche Oberfläche
- Python Code-Editor für das Zeichnen der Aufgaben mittels Python-Code
- Basierend auf brython und turtle graphics
  - Logik der Webanwendung auf Basis von brython anstatt JavaScript
  - Visualisierung des Ergebnisses mit turtle graphics

## 4 Entwurf (grafisch)

## 5 Anforderungen

### 5.1 Funktionale Anforderungen

#### Anforderung 1 - Code-Editor

- Eingeben von Python-Code
- Logik-Highlighting für die Logik in verschiedene Farben (Kann-Ziel)

#### Anforderung 2 - Visualisierung der Ergebnisse

- Ausführen des Codes durch Klick auf Button zeichnet, im Code-Editor definierte Formen auf freie Fläche
- Debugging des Codes durch Klicken auf Button lässt Zeilen einzeln ausführen (Kann-Ziel)

#### Anforderung 3 - Levelwahl / LevelUp

- freie Levelwahl
- LevelUp mit Überprüfung der Programmierlogik (Soll-Ziel)
- LevelUp nach Wissensabfrage zu dem im Level erlernten Programmierkonzept (Soll-Ziel)
  - sollte überspringbar sein

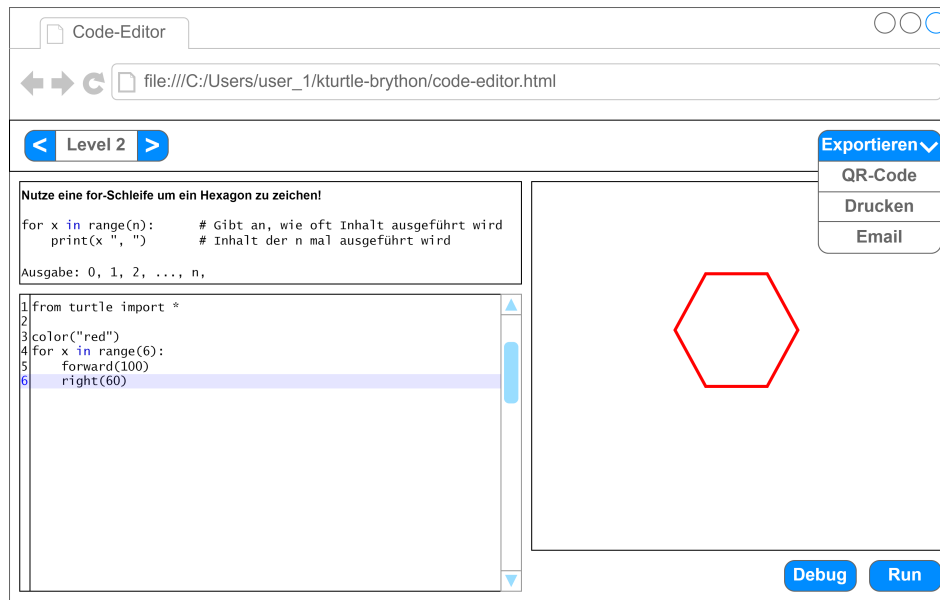


Abbildung 1: grafische Entwurf der Webanwendung

#### Anforderung 4 - Aufgabenstellung

- Darstellung der Aufgabe als Text
- Darstellung von Hinweisen als Code
- Musterlösung der Aufgabe (Soll-Ziel)
  - Passwortgeschützt oder versteckter Button

#### Anforderung 5 - Handout für Schüler

- Drucken des Codes und der bemalten grafischen Oberfläche (Soll-Ziel)
- Hochladen des Codes und der bemalten grafischen Oberfläche auf File-Hosting-System. Freigeben an Schüler als Download-Link in Form von QR-Code (Soll-Ziel)
- Senden des Codes und bemalten grafischen Oberfläche per (Kann-Ziel)

## 5.2 Nichtfunktionale Anforderungen

### Qualitative Anforderungen

- Skalierbarkeit zur einfachen Erweiterung der Level bzw. der Umgebung
- Wartbarkeit

### Rahmenbedingungen

- Kompatibilität auf allen gängigen Browsern
- Funktionstüchtig auf leistungsschwachen PCs

### Gesetzliche Anforderungen

- Nur lizenzfreie Technologie zur Verwendung

## 6 Lieferung der Anwendung

### Lieferumfang

- Ausführbare Webanwendung (nur Frontend)
- Shell-Scripts für Initialisierung der Anwendung (und Reset-Funktion)
- Anleitung (PDF) für ...
  - Schüler
  - Lehrer
  - Erweiterung der Level und Funktionalitäten

### Liefertermin

- vor Weihnachten