

Simulationsprojekt Dokumentation

22.01.2024

Jannick Bath

Inhaltsverzeichnis

1	Projektziel	3
2	Art des Projektes	3
3	Betriebliche Umfeldbeschreibung	3
3.1	Auftragnehmer	3
3.2	Auftraggeber	3
4	Ausgangssituation	3
5	Analyse vom Ist-Zustand	4
6	Soll-Konzept	4
6.1	Architektur und Design	4
6.2	Auswahl der Technologien	4
6.2.1	Frontend	4
6.2.2	Backend	4
6.2.3	Testing	4
6.3	Datenmodellierung	5
6.4	Implementierung der Simulation	5
6.4.1	UI-Design und -Entwicklung	5
6.4.2	Grundmechaniken des Rennspiels	5
6.4.3	Integration des Zufallsfaktors	5
6.5	Benutzerinteraktion und Eingabe-Parameter	6
6.5.1	Benutzerinteraktion	6
6.5.2	Rennstrecken-Ansicht	6
6.5.3	Geschwindigkeitsregler	6
6.5.4	Uhr	6
6.6	Teststrategie	6
6.6.1	End-to-End-Tests für die Gesamtsimulation	6
6.7	Dokumentation	6
7	Ressourcenplanung	7
7.1	Personalplanung	7
7.2	Sachmittelplanung	7
7.3	Kostenplanung	7
8	Ablaufplanung	7
8.1	Planungsphase	7
8.2	Umsetzungsphase	7
8.3	Dokumentationsphase	7

9 Darstellung der Prozessketten	8
9.1 Benutzeroberfläche (UI)	8
9.2 Textbasiertes Rennspiel	8
9.3 Zufallsfaktoren und Items	8
10 Vorgehensweise	8
11 Auftragsergebnis (Soll - Ist Vergleich)	9
11.1 Soll	9
11.2 Ist	9
11.3 Vergleich	9

1 Projektziel

Das Hauptziel dieses Projekts besteht darin, ein herausforderndes Singleplayer-Rennspiel zu entwickeln, bei dem die Spieler ihre Fahrzeuge durch das Schreiben von Texten steuern. Die Schreibgeschwindigkeit beeinflusst die Fortbewegungsgeschwindigkeit des Fahrzeugs direkt. Zusätzlich soll ein spezieller Zufallsfaktor in Form von zufälligen Geschwindigkeitsänderungen für Bots und verschiedenen Items, wie Barrieren, das Spielerlebnis variabel gestalten.

2 Art des Projektes

Das Projekt fällt in die Kategorie der Unterhaltungsspiele und kombiniert Elemente des Rennspiels mit einer Schreibmechanik. Es ist als Singleplayer-Erfahrung konzipiert, bei der Spieler gegen Bots antreten, um als Erster die Ziellinie zu überqueren.

3 Betriebliche Umfeldbeschreibung

3.1 Auftragnehmer

Ich agiere als Auftragnehmer für dieses Projekt im Rahmen meiner schulischen Ausbildung zum Fachinformatiker für Anwendungsentwicklung. Meine Expertise umfasst den Einsatz des Contao Frameworks, PHP, Typescript, JavaScript, SCSS, CSS und HTML. Ich experimentiere auch gerne mit Frontend-Frameworks wie React oder Angular. Dieses Simulationsprojekt ist ein integraler Bestandteil meiner schulischen Ausbildung und bietet eine praxisnahe Anwendung meiner erworbenen Fähigkeiten.

3.2 Auftraggeber

Die Berufsschule BS-Technik Rostock fungiert als Auftraggeber dieses Projekts im Rahmen meiner Ausbildung. Die Anforderungen wurden in Absprache mit der Berufsschule festgelegt, wobei der Fokus auf den folgenden Kriterien liegt:

- Die Oberfläche sollte mindestens 5 Eingabe-Parameter aufweisen.
- Es sollten zwei Zufallsparameter implementiert werden, einmal basierend auf der Gaußschen Normalverteilung und einmal auf der Normalverteilung.
- Es sollten drei verschiedene Stufen der Simulationsgeschwindigkeit vorhanden sein: langsam, mittel, schnell.
- Mindestens eine Animation muss in der Simulation integriert werden. Die Simulation sollte zeitabhängig sein.

4 Ausgangssituation

Die Berufsschule BS-Technik Rostock beauftragt im Rahmen der Ausbildung zum Fachinformatiker für Anwendungsentwicklung die Entwicklung einer Simulation. Die genauen Anforderungen sind bereits unter dem Punkt Auftraggeber formuliert.

5 Analyse vom Ist-Zustand

Der aktuelle Ist-Zustand zeigt, dass bisher keine vorhandene Simulation oder vergleichbare Anwendung mit den spezifischen Anforderungen der Berufsschule BS-Technik Rostock existiert. Dies eröffnet die Möglichkeit, eine Simulation von Grund auf zu entwickeln, die den gewünschten Kriterien entspricht.

6 Soll-Konzept

6.1 Architektur und Design

Für die Architektur des Projekts habe ich die Umsetzung als React-Typescript-Projekt vorgesehen, wobei das Zustandsmanagement durch die Verwendung der Bibliothek SZustandrealisiert wird. Die Architektur folgt einer objektorientierten Herangehensweise, um die Prinzipien der Clean-Code-Ideologie bestmöglich umzusetzen.

Das Design des Projekts konzentriert sich auf eine benutzerfreundliche Gestaltung der Benutzeroberfläche. Dabei liegt der Fokus darauf, die Eingabe-Parameter, Zufallsparameter, Geschwindigkeitsstufen, Animationen und zeitliche Abhängigkeit übersichtlich und intuitiv zu präsentieren. Die Visualisierung der Spielmechaniken erfolgt klar und ansprechend, um eine gute Benutzererfahrung zu gewährleisten.

Der Ablaufplan für die Umsetzung sieht vor, zunächst das UI zu designen und zu entwickeln. Anschließend wird die Implementierung der Grundmechaniken eines textbasierten Rennspiels folgen. Zum Abschluss ist geplant, den Zufallsfaktor durch die Integration bestimmter Items einzuführen.

Des Weiteren ist die Implementierung von Black-Box-Tests vorgesehen, die mithilfe der Bibliothek Cypress durchgeführt werden. Dies stellt sicher, dass die Funktionalitäten des Projekts in verschiedenen Szenarien erfolgreich getestet werden.

6.2 Auswahl der Technologien

Für die Auswahl der Technologien habe ich mich auf bewährte Werkzeuge konzentriert, die die Effizienz und Qualität des Projekts sicherstellen.

6.2.1 Frontend

Die Verwendung von React in Verbindung mit Typescript bietet eine robuste Basis für die Frontend-Entwicklung. Das Zustandsmanagement erfolgt durch die Bibliothek SZustand, was eine effektive Verwaltung des Anwendungsstatus ermöglicht.

6.2.2 Backend

Es war keine Umsetzung eines Backends angedacht.

6.2.3 Testing

Für das Testen habe ich auf Black-Box-Tests vorgesehen, die mithilfe der Bibliothek Cypress durchgeführt werden sollen. Dies ermöglicht umfassende Tests der Benutzeroberfläche und der Anwendungs-

logik in verschiedenen Szenarien.

Die Entscheidungen bei der Auswahl dieser Technologien basieren auf deren Zuverlässigkeit, Leistungsfähigkeit und ihrer Anpassungsfähigkeit an die Anforderungen des Projekts.

6.3 Datenmodellierung

Die Datenmodellierung erfolgt gemäß der Vorgaben von 'Zustand', wodurch eine temporäre Speicherung der Daten in klar abgegrenzten Slices erfolgt. Diese Slices ermöglichen eine klare Strukturierung und Abgrenzung der Daten im Bezug auf verschiedene Komponenten der Simulation.

Da das Projekt kein Backend vorsieht, werden keine persistenten Daten verwendet. Die Datenverwaltung konzentriert sich auf die effiziente Nutzung des Zustand-Managements, um eine konsistente und reaktive Verwaltung der Daten während der Laufzeit der Anwendung zu gewährleisten.

Die objektorientierte Herangehensweise wird dabei beibehalten, um die Struktur und Wartbarkeit der Datenmodelle zu optimieren.

6.4 Implementierung der Simulation

Die Implementierung der Simulation folgt einem strukturierten Ablaufplan, der sicherstellt, dass die Funktionalitäten schrittweise und konsistent entwickelt werden.

6.4.1 UI-Design und -Entwicklung

Die erste Priorität liegt auf der Gestaltung und Entwicklung einer benutzerfreundlichen Benutzeroberfläche. Hierbei werden die Eingabe-Parameter, Zufallsparameter, Geschwindigkeitsstufen, Animationen und die zeitliche Abhängigkeit in die Oberfläche integriert. Die klare und intuitive Darstellung dieser Elemente steht im Mittelpunkt dieses Schrittes.

6.4.2 Grundmechaniken des Rennspiels

Im zweiten Schritt erfolgt die Programmierung der grundlegenden Mechaniken des textbasierten Rennspiels. Hierbei wird die Steuerung durch Tastatureingaben implementiert, um die Fortbewegung des virtuellen Fahrzeugs zu ermöglichen. Der Fokus liegt auf der Umsetzung einer realistischen und ansprechenden Spielerfahrung.

6.4.3 Integration des Zufallsfaktors

Der dritte Schritt beinhaltet die Einbindung von Items, die den Zufallsfaktor in die Simulation bringen. Besonderes Augenmerk wird dabei auf Barrieren als Items gelegt, die den Fortschritt des Spielers verlangsamen. Die Implementierung erfolgt so, dass der Zufallsfaktor das Spielerlebnis bereichert und gleichzeitig herausfordernde Elemente hinzufügt.

Die Umsetzung erfolgt iterativ, wobei nach jedem Schritt eine Überprüfung auf Funktionalität und Benutzerfreundlichkeit erfolgt. Dadurch wird sichergestellt, dass die Simulation schrittweise und konsistent entwickelt wird.

6.5 Benutzerinteraktion und Eingabe-Parameter

Die Umsetzung erfolgt mit dem Fokus darauf, die Nutzerfreundlichkeit zu maximieren und dem Spieler eine transparente Steuerung über alle relevanten Spielaspekte zu bieten.

6.5.1 Benutzerinteraktion

Die zentrale Schnittstelle des Spiels ist die Textbox, in der der Nutzer den Text eingibt, um sein Fahrzeug voranzubringen. Dabei wird direktes Feedback in Echtzeit gegeben, ob das auf der Tastatur getippte Zeichen korrekt ist oder nicht. Dies ermöglicht dem Spieler eine unmittelbare Kontrolle über die Geschwindigkeit seines Fahrzeugs.

6.5.2 Rennstrecken-Ansicht

Unterhalb der Textbox befindet sich die Rennstrecken-Ansicht, die einen Überblick über das eigene Fahrzeug und die anderen Fahrzeuge der Bots bietet. Hier kann der Spieler den Fortschritt zu Ziellinie verfolgen. Die Fahrzeuge werden animiert, um Veränderungen im Fortschritt anschaulich darzustellen und die Spielatmosphäre zu intensivieren.

6.5.3 Geschwindigkeitsregler

Ein zusätzlicher Regler wird implementiert, der es dem Spieler ermöglicht, die Simulationssgeschwindigkeit nach seinen Präferenzen einzustellen. Dies bietet eine flexible Anpassung des Spieltempos.

6.5.4 Uhr

Zur Verbesserung der Immersion wird eine Uhr oder ein Timer integriert, der die aktuelle Simulationszeit anzeigt. Dies gibt dem Spieler einen klaren Überblick darüber, wie lange die Simulation bereits läuft.

6.6 Teststrategie

Die folgende Teststrategie zielt darauf ab, eine robuste und fehlerfreie Simulation zu gewährleisten.

6.6.1 End-to-End-Tests für die Gesamtsimulation

Die automatisierten E2E-Tests fokussieren sich darauf, die Wechselwirkungen zwischen den verschiedenen Komponenten sicherzustellen. Dabei wird die Gesamtsimulation auf verschiedene Szenarien und Nutzereingaben getestet, um eine reibungslose Zusammenarbeit aller Elemente zu gewährleisten.

6.7 Dokumentation

Die Dokumentation erfolgt durch sorgfältige Kommentierung des Codes, wobei besonderes Augenmerk auf die Erläuterung von Funktionen, Strukturen und Implementierungsdetails gelegt wird.

Die Kommentierung zielt darauf ab, die Klarheit und Verständlichkeit der Codebasis zu verbessern.

7 Ressourcenplanung

7.1 Personalplanung

Die alleinige Umsetzung des Projekts erfolgt im Rahmen meiner Ausbildung als Fachinformatiker für Anwendungsentwicklung. Alle Entwicklungsaktivitäten, von der Planung über die Umsetzung bis zur Dokumentation, fallen in meinen Verantwortungsbereich.

7.2 Sachmittelplanung

Die genutzten Tools für die Entwicklung, wie React, TypeScript, Zustand und Cypress, sind kostenfrei verfügbar und erfordern keine zusätzlichen finanziellen Aufwendungen. Diese Auswahl gewährleistet eine kosteneffiziente Entwicklung.

7.3 Kostenplanung

Da alle genutzten Tools kostenfrei sind und keine externen Ressourcen eingebunden werden, entstehen keine direkten Kosten im Rahmen der Entwicklung. Die Kostenplanung beschränkt sich somit auf die effiziente Nutzung der verfügbaren Zeit.

8 Ablaufplanung

8.1 Planungsphase

- Definition der Projektziele und Anforderungen.
- Auswahl der geeigneten Tools und Technologien.
- Festlegung der Architektur und Designgrundlagen.

8.2 Umsetzungsphase

- Design und Entwicklung der Benutzeroberfläche (UI).
- Implementierung der grundlegenden Mechaniken des Text-basierten Rennspiels.
- Integration von Simulationsgeschwindigkeiten und Animationen.
- Einbindung von Items und Zufallsfaktoren.

8.3 Dokumentationsphase

- Kommentierung des Codes für eine klare Nachvollziehbarkeit.
- Erstellung der Benutzerdokumentation.
- Überarbeitung und Verbesserung der Dokumentation nach Bedarf.

9 Darstellung der Prozessketten

9.1 Benutzeroberfläche (UI)

Die Nutzerschnittstelle bildet den Ausgangspunkt der Prozessketten. Hier interagiert der Nutzer mit der Anwendung, gibt Texte ein und erhält Feedback zum Fortschritt. Die Prozesskette beginnt mit der Erfassung der Nutzereingabe in der Textbox und setzt sich fort mit der Verarbeitung dieser Eingabe, um die Simulation entsprechend zu beeinflussen.

9.2 Textbasiertes Rennspiel

Die Mechaniken des textbasierten Rennspiels bilden den zentralen Prozess. Hier werden die eingegebenen Texte verarbeitet, die Simulationsgeschwindigkeiten berechnet und die Animationen der Fahrzeuge gesteuert. Die Schnittstellen zwischen der Benutzeroberfläche und dem Spielprozess sind entscheidend, um eine reibungslose Interaktion zu gewährleisten.

9.3 Zufallsfaktoren und Items

Die Integration von Zufallsfaktoren und Items führt zu weiteren Prozessketten. Hier werden die verschiedenen Elemente, die das Spiel dynamisch gestalten, in den Ablauf eingebunden. Die Schnittstellen zwischen dem textbasierten Rennspiel und den Zufallsfaktoren sind dabei essenziell, um einen organischen und unterhaltsamen Spielverlauf zu ermöglichen.

10 Vorgehensweise

Die Entwicklung des Simulationsprojekts orientierte sich an einem Wasserfall-Vorgehensmodell. In der Planungsphase wurden die Projektziele konkretisiert, indem die Idee eines textbasierten Rennspiels formuliert wurde, das sowohl die Anforderungen des Simulationsprojekts erfüllt als auch unterhaltsamen Spielspaß bietet.

Als Methode zur Visualisierung und groben Konzeptualisierung diente das Design-Tool Figma, welches die Grundstruktur des Projekts skizzierte. Die genauen Funktionen wurden zu diesem Zeitpunkt noch nicht im Detail festgelegt.

Während der Umsetzungsphase traten Herausforderungen auf, insbesondere im Bereich des State-Managements. Die Standard-Features von Reacts State-Management erwiesen sich aufgrund zahlreicher Abhängigkeiten zwischen verschiedenen Komponenten als unzureichend. Als Lösung wurde die Integration der Zustand-Bibliothek gewählt, um eine effiziente Verwaltung des Anwendungsstatus zu ermöglichen.

Es gab Anpassungen an der ursprünglichen Planung, insbesondere im Bereich des State-Managements. Die anfängliche Absicht, Redux zu verwenden, wurde aufgrund von Lernschwierigkeiten durch Zustand ersetzt, um einen reibungslosen Entwicklungsprozess sicherzustellen.

Die Tests wurden als End-to-End (E2E) Tests mithilfe der Cypress-Bibliothek durchgeführt.

Der Fokus lag dabei zunächst auf der Überprüfung der Kernfunktionalitäten, einschließlich der Texteingabe, Berechnung von Geschwindigkeiten und Steuerung der Animationen.

11 Auftragsergebnis (Soll - Ist Vergleich)

11.1 Soll

Die ursprüngliche Vision des Projekts war die Entwicklung eines textbasierten Rennspiels, das den Anforderungen des Simulationsprojekts entspricht und gleichzeitig ein unterhaltsames Spielerlebnis bietet. Die spezifischen Kriterien umfassten die Implementierung von Eingabeparametern, Zufallsparametern, verschiedenen Simulationsgeschwindigkeiten, einer Animation und einer zeitabhängigen Simulation.

11.2 Ist

Das resultierende Projekt ist ein funktionsfähiges textbasiertes Rennspiel, das erfolgreich die definierten Kriterien des Simulationsprojekts erfüllt. Die Simulation ist zeitabhängig und zusätzliche Elemente wie Barrieren sorgen für eine vielseitige und herausfordernde Spielerfahrung.

11.3 Vergleich

Der Vergleich zeigt eine weitgehende Übereinstimmung zwischen den ursprünglichen Zielen und dem erreichten Ergebnis. Die Funktionalitäten des Spiels erfüllen die gestellten Anforderungen und bieten darüber hinaus einen zufälligen und unterhaltsamen Aspekt durch die Implementierung von Items und Hindernissen.