

Pflichtenheft: Game of Life – Projekt

Einleitung:

Das Ziel ist die Entwicklung einer plattformunabhängigen Webanwendung, die eine erweiterbare um individuelle Regeln erweiterte Version des klassischen Game of Life implementiert.

Dabei sollen zusätzlich Anpassungen und Erweiterungen ermöglicht werden, um individuelle Regeln umzusetzen. Das Endergebnis soll eine benutzerfreundliche Webanwendung sein, die es den Benutzern ermöglicht, das Game of Life nach Standardregeln zu spielen oder eigene Regeln zu definieren und das Spiel entsprechend anzupassen. Das Design und die Einführung ins Spiel sollen so ansprechend sein, dass es auch für Personen ohne Vorwissen verständlich ist.

Auftrag:

Der Auftrag umfasst die Konzeption, Entwicklung und Implementierung der Webanwendung. Zu den Meilensteinen gehören die Erstellung eines Designs, die Implementierung der standardmäßigen Game of Life Regeln, die Benutzeroberfläche, die Möglichkeit eigene Regeln zu definieren, sowie die Fehlerbehebung und Qualitätssicherung. Das Projekt beinhaltet die Vorgabe, dass die Webanwendung plattformunabhängig und responsiv ist. Spezielle Besonderheiten des Projekts beinhalten die Möglichkeit zur Erweiterung um benutzerdefinierte Regeln und die Implementierung von Speicher- und Ladeoptionen für Spielstände.

Die Spieler haben 2 Status:

- Pixel, die der Spieler platzieren kann.
- Health, den der Spieler verliert, wenn er keine Pixel mehr hat oder gegen die er Pixel eintauschen kann.

In der Mitte ist ein Pixel markiert, der den Spieler repräsentiert. Sollte dieser Pixel durch Bewegung sterben, wird ein zufälliger Pixel als Nachbar erstellt, der als neuer Spieler neu gewählt wird. Sollte der Spieler Pixel sterben, ohne dass ein neuer Nachbar in seiner Nähe entsteht, verliert der Spieler wieder ein Leben.

Der Spieler kann Pixel um sich herum erstellen oder zerstören. Wenn er Pixel erstellt, werden diese von seinen eigenen Pixeln abgezogen, und wenn er Pixel zerstört, erhält er zusätzliche Pixel.

Er kann auch Pixel erhalten, wenn er auf zufällig erstellte Pixel in der Karte trifft oder ein Pixel, dass der Spieler erstellt hat, sich in der Nähe des zufällig erstellten Pixels aufhält (direkte Nachbarschaft).

Das Ziel des Spielers ist es, Zonen auf der Karte einzunehmen, indem er eine bestimmte Anzahl an selbst erstellten Pixeln in der Zone behält.

Der Spieler kann das Spiel jederzeit pausieren, um Pixel zu platzieren oder zu zerstören. Das gesamte Spielfeld wird dann eingefroren.

Auf der Seite kann er ein Menü öffnen, das Strukturen enthält, die der Spieler mit Hilfe seiner Pixel erschaffen kann.

Bereits bestehende Systeme oder Produkte:

Es sind keine bereits bestehenden, projektbezogenen Systeme oder Produkte bekannt, die vom Auftraggeber erledigt wurden und nicht mehr vom Auftragnehmer umgesetzt werden müssen.

Teams und Schnittstellen:

Das Entwicklerteam besteht aus zwei leitenden Entwicklern: Noel und Michael, sowie zwei Designern: Marcel und Fabian.

Arnis ist Scrum-Master und hilft aufgrund seiner Erfahrung in beiden Bereichen bei beiden Teams mit und ist für das Dokumentieren zuständig.

Schnittstellen zu anderen Abteilungen oder externen Partnern sind derzeit nicht vorgesehen und werden voraussichtlich nicht benötigt. Die Kommunikation erfolgt hauptsächlich über wöchentliche Besprechungen sowie über unsere Discord Gruppe.

Rahmenbedingungen: Die Gesamtbearbeitungszeit des Projekts beträgt bis zu zehn Wochen. Die geplanten Betriebs- und Arbeitszeiten entsprechen den regulären Arbeitszeiten des Auftragnehmers. Unser Gesamtes Projektmanagement findet auf GitHub statt und ist dort in Arbeitspaketen verschieden Arbeitsansichten und Meilensteinen aufgeteilt.

Technische Anforderungen: Die Umsetzung erfordert die Verwendung von gängigen Webtechnologien wie HTML, CSS und TypeScript. Darüber hinaus wird ein serverseitiges Framework für die Handhabung von Benutzereingaben und Datenpersistenz verwendet. Es wird erwartet, dass das Entwicklungsteam über ausreichende Rechenleistung, Entwicklungsumgebungen und Testgeräte verfügt, um die Anwendung plattformunabhängig zu testen. Der Spielstand wird innerhalb

des Browsers gespeichert, somit wird keine Datenbank benötigt, dies hilft auch unseren unser Co2 Fußabdruck zu verringern.

Problemanalyse: Zu erwartende Probleme könnten die komplexe Logik für die individuellen Regeln sowie die Speicherung und Wiederherstellung von Spielständen sein. Um diesen Problemen zu begegnen, wird eine eingehende Testphase und rechtzeitige Fehlerbehebung eingeplant.

Qualität: Die Anwendung muss sicherstellen, dass sie plattformunabhängig funktioniert und eine intuitive Benutzererfahrung bietet. Die Qualitätssicherung erfolgt durch umfassende Tests und Code-Reviews. Die Abnahme erfolgt nach definierten Kriterien mit dem Auftraggeber.

Projektentwicklung: Die Projektentwicklung ist in verschiedene Phasen unterteilt, einschließlich Planung, Design, Implementierung, Tests und Bereitstellung. Alle Phasen werden in regelmäßigen Abständen mit dem Auftraggeber abgestimmt, um sicherzustellen, dass die Ziele und Erwartungen erfüllt werden.