ILA3\_0110: Projektdokumentation

Niklaus Andrea, Zumstein David

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
| 17.08.22 | 0.1.0 | Erste Version | Niklaus |
| 24.08.22 | 0.2.0 | Zweite Version | Niklaus |
| 31.08.23 | 0.3.0 | Dritte Version | Niklaus |
| 07.09.23 | 0.4.0 | Vierte Version | Niklaus |
| 14.09.23 | 0.5.0 | Fünfte Version | Niklaus |
| 21.09.23 | 0.6.0 | Sechste Version | Zumstein |
| 28.09.23 | 1.0.0 | Projektabschluss |  |

# Informieren

## Ihr Projekt

Unser Projekt ist ein Tic Tac Toe-Game geschrieben mit F#. Unser Ziel ist es damit unsere Kenntnisse in F# zu erweitern. Das Game ist für die Konsolenanwendung in Visual Studio geschrieben.

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | Muss | Funktional | Es gibt ein 3 x 3 Feld |
| 2 | Muss | Funktional | Es existieren 2 Spieler |
| 3 | Muss | Rand | Das Programm ist in F# geschrieben |
| 4 | Muss | Rand | Das Programm ist als eine Konsolenanwendung in Visual Studio erstellt. |
| 5 | Muss | Funktional | 3 in einer Reihe führen zu einem Sieg |
| 6 | Muss | Funktional | Alle Felder ausgefühlt ohne 3 gleiche Zeichen in einer Reihe führt zu Unentschieden |
| 7 | Muss | Funktional | Fehleingaben werden gefangen |
| 8 | Muss | Funktional | Man kann auswählen an welchem Ort man sein Zeichen platzieren will |
| 9 | Kann | Funktional | Das Programm wird nach einem Spiel geschlossen |

## Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Voraussetzung** | **Eingabe** | **Erwartete Ausgabe** |
| 1.1 | Programm gestartet | Auswahl vom Ort des Zeichens | An dem Ort wird das Zeichen platziert |
| 2.1 | Programm gestartet | - | Spielfeld wird angezeigt |
| 3.1 | Programm gestartet | Player 1 hat ausgewählt | Player 2 kann auswählen |
| 4.1 | Schon 2 gleiche Zeichen nebeneinander platziert | 3tes gleiche Zeichen neben den 2 platzieren | "Glückwunsch, Spieler - hat gewonnen" |
| 5.1 | Alle Felder bis auf eins ausgefüllt | letztes Feld ausfüllen | "Unentschieden" |
| 6.1 | Spiel gespielt | Enter-Taste | Programm schliesst sich |

1. Planen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** |
| 1.1 | 21.09.2023 | Spieler erstellen | 45 Minuten |
| 2.1 | 21.09.2023 | Funktionierendes Spielfeld Erstellen | 240 Minuten |
| 3.1 | 21.09.2023 | Gewinnfunktion | 300 Minuten |
| 4.1 | 21.09.2023 | Spieler wechseln | 60 Minuten |
| 5.1 | 21.09.2023 | Fehleingaben abfangen | 45 Minuten |
| 6.1 | 21.09.2023 | Eingabeaufforderung | 45 Minuten |
| 7.1 | 21.09.2023 | Spiellogik | 700 Minuten |
| **Total:** |  |  | **1'435 Minuten** |

1. Entscheiden

Das Projekt wird als Konsolenanwendung mit F# geschrieben.

1. Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** |
| 1.1 | 21.09.2023 | Spieler erstellen | 45m | 30m |
| 2.1 | 21.09.2023 | Funktionierendes Spielfeld Erstellen | 240m | 240m |
| 3.1 | 21.09.2023 | Gewinnfunktion | 300m | 270m |
| 4.1 | 21.09.2023 | Spieler wechseln | 60m | 50m |
| 5.1 | 21.09.2023 | Fehleingaben abfangen | 45m | 60m |
| 6.1 | 21.09.2023 | Eingabeaufforderung | 45m | 45m |
| 7.1 | 21.09.2023 | Spiellogik | 700m | 700m |
| **Total:** |  |  | 1'435 | **1'395 Minuten** |

1. Kontrollieren

## Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| 1.1 | 21.09.2023 | OK | Zumstein |
| 2.1 | 21.092023 | OK | Zumstein |
| 3.1 | 21.09.2023 | OK | Zumstein |
| 4.1 | 21.09.2023 | OK | Zumstein |
| 5.1 | 21.09.2023 | OK | Zumstein |
| 6.1 | 21.09.2023 | OK | Zumstein |

Fazit: Unser Programm ist Funktionstüchtig.

1. Auswerten

Unser Projekt erfüllt alles was es muss und könnte so dem Kunden gegeben werden. Verbesserungsmöglichkeiten sind jedoch da.