**ProvCasino**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
| 22.02.2024 | 0.1.0 | Erste Version | Zumstein |
| 29.02.2024 | 0.2.0 | Zweite Version | Zumstein |
| 07.03.2024 | 0.3.0 | Finale Version | Zumstein |

## Mein Projekt

Ich mache eine SlotMachine Applikation mit C#. Gleichzeitig will ich ein Pattern einbauen um dies zu repetieren.

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | Muss | Funktional | Das Programm generiert bei der Slotmachine eine zufällige Reihenfolge an 3 Zahlen |
| 2 | Muss | Funktional | Wenn die drei Zahlen bei der Slotmachine gleich sind, gewinnt der Spieler |
| 3 | Muss | Funktional | Wenn die drei Zahlen bei der Slotmachine nicht gleich sind, verliert der Spieler |
| 4 | Muss | Qualität | Das Programm hat eine Art von Pattern eingebaut |
| 5 | Muss | Funktional | Man kann das Programm beenden |
| 6 | Kann | Funktional | Man kann erneut spielen |
| 7 | Muss | Rand | Die Applikation ist mit C# geschrieben |
| 8 | Muss | Rand | Die Applikation ist auf GitHub zu finden |

## Testfälle

|  |  |  |
| --- | --- | --- |
| **Nummer** | **Eingabe** | **Erwartete Ausgabe** |
| 1.1 | Der Benutzer startet die Applikation | Das Programm generiert eine zufällige Reihenfolge an 3 Zahlen |
| 2.1 | Die 3 Zahlen sind gleich | "Wow! sie haben gewonnen:)" |
| 3.1 | Die 3 Zahlen sind nicht gleich | "Schade Schokolade" |
| 4.1 | Nach einer Runde "exit" schreiben | Spiel wird beendet |
| 5.1 | Nach eine Runde Enter drücken | Spiel startet erneut |

## Planen

|  |  |  |
| --- | --- | --- |
| **Nummer** | **Frist** | **Zeit (geplant)** |
| 1.1 | 07.03 | 120 Minuten |
| 2.1 | 07.03 | 60 Minuten |
| 3.1 | 07.03 | 20 Minuten |
| 4.1 | 07.03 | 250 Minuten |
| 5.1 | 07.03 | 45 Minuten |
| 6.1 | 07.03 | 45 Minuten |
| **Total:** |  | **540 Minuten** |

## Entscheiden

Ich habe mich für das Observer Pattern entschieden, da ich es schon einmal in einem anderen Projekt benutzt habe.

## Realisieren

|  |  |  |  |
| --- | --- | --- | --- |
| Nummer | Frist | Zeit (geplant) | Zeit (effektiv) |
| 1.1 | 07.03 | 120 Minuten | 100 Minuten |
| 2.1 | 07.03 | 60 Minuten | 60 Minuten |
| 3.1 | 07.03 | 20 Minuten | 10 Minuten |
| 4.1 | 07.03 | 250 Minuten | 240 Minuten |
| 5.1 | 07.03 | 45 Minuten | 30 Minuten |
| 6.1 | 07.03 | 45 Minuten | 30 Minuten |
| **Total:** |  | **540 Minuten** | **470 Minuten** |

## Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| 1.1 | 07.03.24 | **OK** | Zumstein |
| 2.1 | 07.03.24 | **OK** | Zumstein |
| 2.2 | 07.03.24 | **OK** | Zumstein |
| 2.3 | 07.03.24 | **OK** | Zumstein |
| 3.1 | 07.03.24 | **OK** | Zumstein |
| 3.2 | 07.03.24 | **OK** | Zumstein |
| 4.1 | 07.03.24 | **OK** | Zumstein |

## Auswerten

Ich hatte für das Projekt eher wenig Zeit, weswegen ich ganz zufrieden mit dem Endprodukt bin. Zusätzlich konnte ich ein Pattern repetieren um mein vorheriges Wissen aufzufrischen.