

# Lernatelier: Projektdokumentation

Bernegger

Datum	Version	Änderung	Autor
25.8.2021	0.0.1	Protokoll	Bernegger
1.9.2021	0.0.2	Protokoll und Anfang des Codes geschrieben	Bernegger
8.9.2021	0.0.3	Protokoll und weiter codiert und ziel 4 erfüllt	Bernegger
15.9.2021	0.04	Protokoll und try catch	Bernegger
22.9.2021	1.0.0	Finale Version	Bernegger

## 1. Informieren

### 1.1 Ihr Projekt

In meinem Projekt programmiere ich mit C# ein Spiel, in welchem man eine Zahl eingeben kann und der Computer gibt aus, ob die eingegebene Zahl mit der zufällig generierten Zahl übereinstimmt.

### 1.2 Quellen

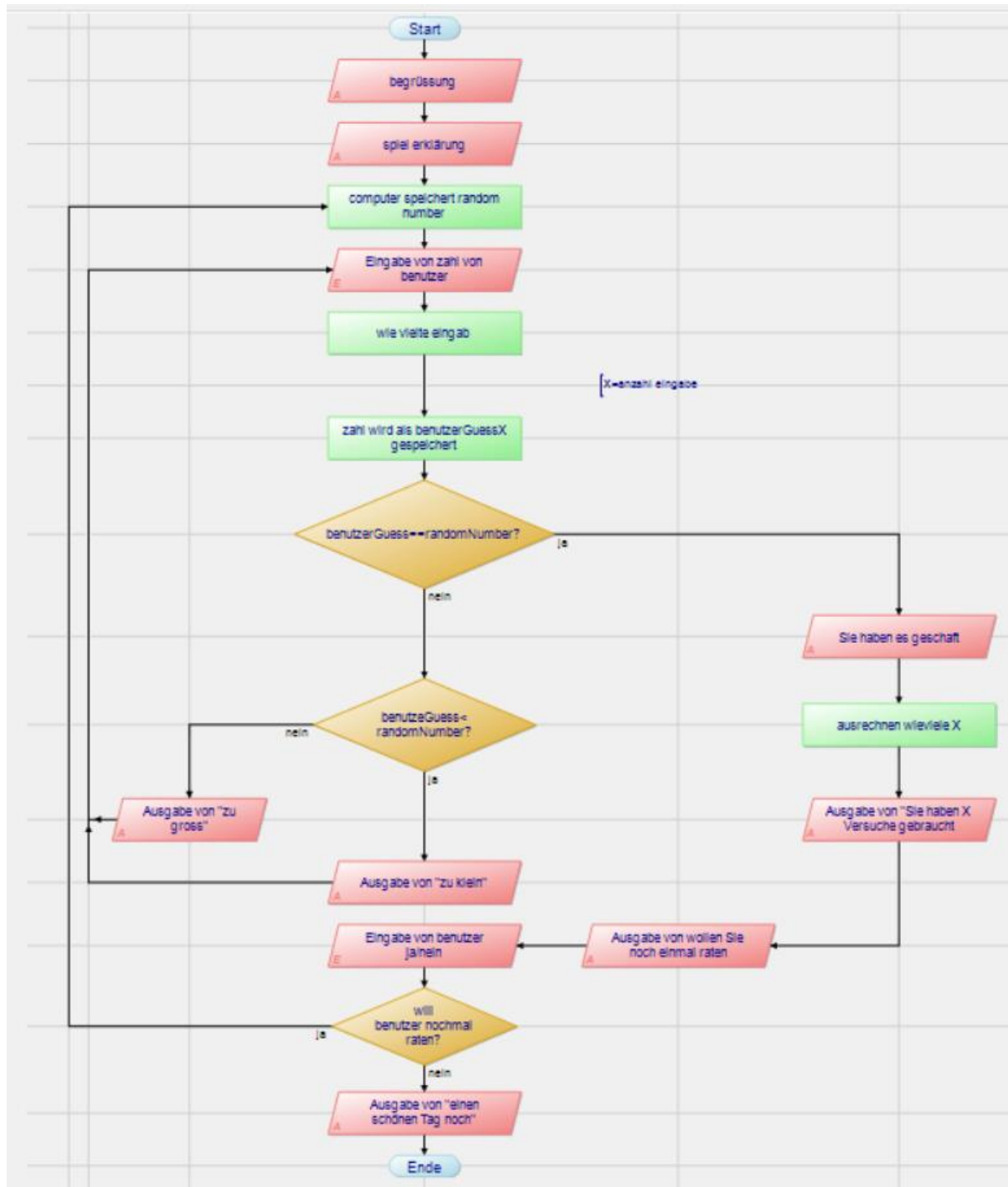
Ppp vom Donnerstag (PR\_390\_Methodenaufrufe).

Freunde von meinem Tisch in der BBB im Zimmer B421.

### 1.3 Anforderungen

Nummer	Muss / Kann?	Funktional? Qualität? Rand?	Beschreibung
1	Muss	Funktional	Der Computer speichert eine Zufallszahl zwischen 1 bis 100 als Geheimzahl.
2	Muss	Funktional	Für jede der geratenen Zahlen gibt der Computer einen Hinweis aus: Die geratene Zahl ist niedriger als die Geheimzahl. Die geratene Zahl ist grösser als die Geheimzahl. Die Geheimzahl wurde erraten.
3	Muss	Funktional	Der Benutzer kann Zahlen raten.
4	Muss	Qualität	Wenn die Geheimzahl erraten wurde, soll die Anzahl der Rateversuche ausgegeben werden.
5	Muss	Qualität	Das Programm soll mit Fehleingaben umgehen oder sie vermeiden können.
6	Kann	Qualität	Emotionen einbringen mit Emojis

## 1.4 Diagramme



## 1.5 Testfälle

---

Nummer	Voraussetzung	Eingabe	Erwartete Ausgabe
1.1	Programm aufstarten Code eintippen	Code von C#	Begrüssung
1.2	Programm starten und im Debugger schauen	Code für Zufallsgenerator	Der Computer merkt sich eine Zahl von 1-100
2.1	Programm starten und im Debugger schauen und bracke Points setzen	Code für ein/Ausgabe	Der Benutzer tippt eine Zahl ein, die dann ausgegeben wird
2.2	Programm starten und im Debugger schauen	Code für richtig/falsch	Der Computer überprüft, ob es richtig oder falsch ist
3.1	Programm starten und im Debugger schauen und bracke Points setzen	Code zu tief/hoch/wahr	Der Computer gibt die geratende Zahl aus
4.1	Programm starten und im Debugger schauen	Code für Ausgabe für Anzahl versuche	Der Computer gibt die Anzahl versuche aus
5.1	Programm starten und bracke Points setzen	Try catch einfügen	Fehlerhafte Eingabe

## 2. Planen

---

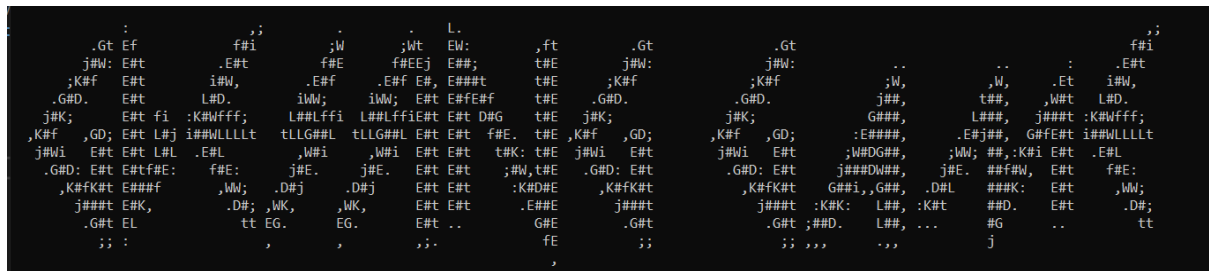
Nummer	Frist	Beschreibung	Zeit (geplant)
1.1	1.9	Programm für eine Zufallszahl schreiben	40min
1.2	1.9	Zahl speichern und verstecken	30min
2.1	1.9	Für jede der geratenen Zahlen gibt der Computer einen Hinweis aus	45min
2.2	8.9	Was Computer machen soll, wenn richtig	30min
3.1	8.9	Der Benutzer kann Zahlen raten.	10min
4.1	8.9	die Anzahl der Rateversuche ausgegeben	20-40min
5.1	15.9	Fehleingaben umgehen oder sie vermeiden können	45-60min
6	15.9	Emojis einfügen und grosser Text aus Zeichen	20min

### 3. Entscheiden

Ich habe mich dazu entschieden Emojis einzubringen, um Emotionen beim Benutzer zu erwecken. Ebenfalls so dass es wenige Monoton wirkt.

Auch habe ich einen Titel eingefügt. |

∨



### 4. Realisieren

Nummer	Frist	Beschreibung	Zeit (geplant)	Zeit (effektiv)
1.1	1.9	Programm für eine Zufallszahl schreiben	40min	35min
1.2	1.9	Zahl speichern und verstecken	30min	20min
2.1	1.9	Für jede der geratenen Zahlen gibt der Computer einen Hinweis aus	45min	40min
2.2	8.9	Was Computer machen soll, wenn richtig & falsch	30min	35min
3.1	8.9	Der Benutzer kann Zahlen raten.	30min	20min
4.1	8.9	die Anzahl der Rateversuche ausgegeben	20-40min	30min
5.1	15.9	Fehleingaben umgehen oder sie vermeiden können	45-60min	55min
6	15.9	Emojis einfügen und grosser Text aus Zeichen	20min	30min

## 5. Kontrollieren

### 5.1 Testprotokoll

Nummer	Datum	Resultat	Durchgeföhrt
1.1	22.9	Ok	Bernegger
1.2	22.9	Ok	Bernegger
2.1	22.9	Ok	Bernegger
2.2	22.9	Ok	Bernegger
3.1	22.9	Ok	Bernegger
4.1	22.9	Ok	Bernegger
5.1	22.9	Ok	Bernegger
6	22.9	Ok	Bernegger

Es hat alles gut funktioniert.

## 6. Auswerten

Das mit dem Designen lief gut und es hat auch Spass gemacht.

Jedoch hatte ich Probleme mit der Fehleingeben Abfangung, da ich es zu allem ersten Mal mit "try catch" versucht hatte.

Windows + L