

Brändi Dog 2.0 Qualitätsmanagement

Ziele

Nach Aufgabenstellung des Projekts müssen *messbare* Ziele für die Qualitätssicherung unseres Programms erforderlich. Uns sind vor Allem die Merkmale **Funktionalität** und **Effizienz** wichtig.

Funktionalität

Als Indikatoren der Funktionalität haben wir uns folgende Kriterien ausgesucht:

Betriebssysteme

Das Programm soll auf den drei grossen Betriebssystemen (Linux, Mac, Windows) laufen, d.h. Sowohl Server als auch Client starten und beenden richtig, der Client arbeitet sowohl im Chat als auch im Spiel richtig.

Der Punkt Betriebssystem wird auf einer Skala von 0 bis 3 bewertet, einen Punkt für jedes drei OS, welches der Anforderung genügt. Es sind nur ganze Punkte vorgesehen.

Client-Server-Verbindung

Der Client verarbeitet und zeigt sämtliche Nachrichten vom und an den Server richtig (keine (De-)Kodierungsfehler). Mit Blick auf den Indikator Betriebssysteme wird die (De-)Kodierung für die drei Systeme separat betrachtet

Bewertet wird hier auch auf einer Skala von 0 bis 3 bewertet. Für jedes System, das keine (De-)Kodierungsfehler aufweist, wird ein Punkt vergeben. Einen halben Punkt kriegt ein System, welches höchstens bei Sonderzeichen Fehler macht. Keinen Punkt kriegt ein System, welches sogar bei normalen Buchstaben, Zahlen und Satzzeichen Fehler macht.

Regeltreues Spiel

Das Spiel soll natürlich die Regeln, welche den Spielern vermittelt werden, auch richtig umsetzen. Die Karten tun, was sie tun sollen, die Murmeln bewegen sich richtig und Gewinner werden richtig festgestellt.

Bewertet wird auch hier auf einer Skala von 0 bis 3. 3 Punkte gibt es, wenn das gesamte Regelwerk richtig funktioniert. Es gibt je einen Punkt Abzug, wenn Spielende bzw. Gewinner nicht oder nicht richtig festgestellt wird, wenn eine oder mehrere der Kartenaktionen nicht richtig implementiert sind und wenn sich die Murmeln nicht entsprechend den Regeln bewegen (z.B. Eine Murmel geht ins Ziel, statt noch eine Runde zu drehen).

Effizienz

Als Indikator für die Effizienz dient die Menge des gebrauchten Arbeitsspeicher.

Arbeitsspeicher

Das Programm soll nicht mehr als ~~1024MB~~ 320MB¹ Arbeitsspeicher pro Instanz verbrauchen.

Bewertet wird hier auf einer Skala von 0 bis 3. 3 Punkte kriegt das Programm, wenn es nicht mehr als 320MB verbraucht. Für jedes weitere 16MB „Paket“, das angefangen wird, wird ein halber Punkt abgezogen, bis beim Überschreiten von 384MB Arbeitsspeicher 0 Punkte erreicht sind.

¹ Ursprünglich waren 1024MB gedacht, doch nach unseren ersten Messungen stellten wir fest, dass diese Schranke (nahezu) unerreichbar hoch angesetzt war, sodass ein Fehlschlag (nahezu) ausgeschlossen gewesen wäre.

Auswertung der Bewertung

In unserer Bewertung sind maximal 12 Punkte zu holen. Glücklicherweise sind wir auch erst mit allen zwölf Punkten. Als genügend, also quasi als absolutes Minimum, betrachten wir neun Punkte.

Ungemessenes Ziel: Benutzbarkeit

Neben diesen messbaren Zielen gibt es auch viele Dinge, die ohne grossen Aufwand nicht messbar sind. Mit dem Wissen, dass wir keine Masse für diese Punkte finden werden, die im kleinen Rahmen des Programmierprojekts noch sinnvoll wären, behalten wir neben Funktionalität und Effizienz trotzdem noch die **Benutzbarkeit** des Programms im Auge:

Intuitives Design

Funktionen in Spiel und Chat sollen an sinnvollen Orten zu finden sein; ein User soll nicht lange nach einem Button oder Menü suchen müssen. Buttons sollen natürlich das tun, was drauf steht.

Wenige Klicks

Aktionen (vor Allem im Spiel) sollen mit möglichst wenigen Klicks ausgeführt werden können. Die meisten Spielzüge brauchen zwei Klicks; jeder weitere muss begründeterweise da sein.

Messungen

Messungen werden mit jedem neuen Build neu durchgeführt. Neben den JUnit Tests, die dank dem Buildscript automatisch weite Teile der Spiellogik überprüfen, sind auch manuelle Tests erforderlich. Die Messung der Effizienz, sprich des Arbeitsspeicherbedarfs, erfolgt über den Task Manager des Betriebssystems. Für die Funktionalität haben wir eine Reihe Tests, die uns zeigen sollen, wo beim Chat und beim Spiel Nachbesserungsbedarf besteht. Zum Testen der Regeltreue wird ausserdem das Spiel einfach nur gespielt; dies hat uns bereits über eine Inkonsistente Regelanwendung informiert. Dies zeigt uns auch, ob das Spiel immer noch auf den drei grossen Betriebssystemen läuft, da in unserer Gruppe alle drei vertreten sind. Die Messungen werden in einem separaten Testlog niedergeschrieben. Dieses dient auch als Referenz zur Bewertung unseres Erfolges.

Die Benutzbarkeit hingegen wird nicht gemessen; stattdessen hinterfragen wir gegenseitig unsere Designentscheidungen innerhalb der Gruppe: Wieso sind zum Spielen dieser Karte vier Klicks nötig, wäre dort ein einfacher Knopf nicht besser als ein Punkt, der im Menü versteckt ist? Lediglich die Beschriftung der einzelnen Knöpfe und Menüs kann einigermaßen objektiv betrachtet werden.

Erfolg (Stand: 22.Mai 2017)

• Funktionalität:	9	
◦ Betriebssysteme:	3/3	
◦ Client-Server-Verbindung	3/3	
◦ Regeltreues Spiel:	3/3	
• Effizienz:	3/3	
◦ Arbeitsspeicher:	2.5/3	TOTAL: 11.5/12

Mit 11.5 von 12 Punkten können wir zufrieden sein, auch wenn wir beim Speicher nur knapp an der vollen Punktezahl entfernt sind. Weiterhin soll beachtet werden, dass die letzten Tests, wegen der sich schnell nähernden Deadline, nicht von ausgiebigen Spielsessionen für vertieftes Bugfinding begleitet wurden. Ein gewisses „Restrisiko“ besteht also.