



Software Engineering Übung 1

Programmverständnis, Dokumentation

1 Informationen

1.1 Daten

- Ausgabe Di 17.09.2013
- Abgabe Mo 30.09.2013 bis 23:59 Uhr
- Besprechung am Di 08.10.2013 um 12:15 Uhr

1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen Ex[n] [NameA Matrikelnummer].pdf abgegeben werden, wobei [n] die Nummer der Übung ist und [NameA Matrikelnummer] der Name und Matrikelnummer des Studenten sind. Die PDF Datei sollte ausserdem ebenfalls Ihren Namen und Matrikelnummer beinhalten.

Bitte senden sie ihre Lösungen via OLAT. Der Betreff der Lösung sollte mit [SE EX HS13] beginnen. Falls Sie zusätzliche Abgabematerialien (z.B. Source Code) haben, sende sie das als Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen Sie das Archiv anhand der oben erwähnten Konventionen.

Diese Übung sollte individuell gelöst werden. Der Student muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

2 ShootTheDuck

2.1 Allgemeines

Diese Aufgabe basiert auf den Java Projekt "*ShootTheDuck*". Das Projekt besteht aus 5 Java Klassen und beinhaltet alle wichtigen Elemente eines modernen Videospiele.

Der Beispielordner beinhaltet die Java Klassen und die Assets/Bilder, die wir in das Spiel eingebunden haben.

Damit alle die gleiche Ausgangslage haben, verwenden Sie bitte den Quellcode, den Sie auf den Webseiten zur Übung finden.

2.2 Installation

In Eclipse, gehen Sie auf *File* und dann *Import*. Im neuen Fenster klicken Sie auf *General* und dann *Existing Projects into Workspace*. Als Archiv wählen Sie die *ShootTheDuck.zip*. Klicken Sie auf *Finish*. Eclipse IDE wird nun ein neues Projekt unter dem Namen *ShootTheDuck* im Workspace anlegen.



Bild 1. Startbild des Spiels

3 Verstehen des Quellcodes (12 Punkte)

In dieser Aufgabe geht es darum, den Quellcode zu verstehen. Hierzu erstellen Sie Modelle, welche die Struktur der Software sowie ihr Verhalten in gewissen Situationen dokumentieren.

3.1 Struktur (3 Punkte)

Dokumentieren Sie die im Projekt vorkommenden Klassen und Schnittstellen (Interfaces) in einem Klassendiagramm. Private und Protected Methoden brauchen Sie nicht zu berücksichtigen.

3.2 Methoden (3 Punkte)

Betrachten Sie die Methoden *mouseButtonState* und *mouseKeyStatus* in der Klasse *Canvas*. Analysieren und erläutern Sie, wofür diese Methoden benutzt werden.

3.3 Verhalten (6 Punkte)

- Beschreiben Sie was geschieht, wenn die linke Maustaste geklickt wird am Anfang des Spieles. Welche Methode wird aufgerufen und wieso? Modellieren Sie den Ablauf mit einem UML-Sequenzdiagramm.
- Modellieren Sie den Ablauf des ersten Aufrufs der Methode *GameLoop()* mit einem UML-Sequenzdiagramm.

4 Nachdokumentation und Änderungen (8 Punkte)

4.1 Programmierstil (1 Punkt)

Lesen Sie das Kapitel zur Benennung von Methoden in den *Entwicklungsrichtlinien für Java Software* (Java Style Guide, JSG), die Sie von den Übungswebseiten herunterladen können. Analysieren Sie nun die im *ShootTheDuck*-Projekt verwendeten Methodennamen. Welchen Fehler erkennen Sie?

4.2 Dokumentation (4 Punkte)

Die Klasse *Game* beinhaltet unter anderem die Methode *UpdateGame()*. Dokumentieren Sie diese Methode nach den Entwicklungsrichtlinien für Java-Software.

4.3 Änderungen im Code (3 Punkte)

Oben in der Mitte des Spielfelds soll ein Cancel-Knopf platziert werden. Wird dieser Knopf gedrückt, so soll das Spiel sofort verlassen werden.

Implementieren Sie diese Änderung und dokumentieren Sie neu geschriebenen oder geänderten Code entsprechend den Entwicklungsrichtlinien für Java-Software.

Tipp: Der Code für die Erzeugung des Knopfes sollte im Konstruktor der Klasse *Window* platziert werden.