



HSB

Hochschule Bremen
City University of Applied Sciences

Fakultät 4
Elektrotechnik und Informatik
Prof. Dr. Ing. Heiko Mosemann
Flughafenallee 10
28199 Bremen

0421-5905-5601
Heiko.Mosemann@hs-bremen.de

Bremen, April 2023

Aufgabenstellung Entwicklungsarbeit GUIPROG im SoSe 23

Liebe Menschen,

die Prüfungsform zum Modul GUIPROG ist die Entwicklungsarbeit. Die Prüfungsordnung (Abschnitt II: Prüfungsformen § 7 Arten der Prüfungsleistungen, Studienleistungen) sagt:

„Eine Entwicklungsarbeit besteht in der Erstellung und Demonstration einer Computer-Software einschließlich der zugehörigen Dokumentation. Die Dokumentation umfasst in der Regel folgende Dokumente:

- die Aufgabenstellung,
- die Anforderungsdefinition,
- den Entwurf,
- das Quellprogramm,
- die Testdokumentation,
- Benutzungshinweise und
- ein Anwendungsbeispiel.

Die Prüfungsleistung kann als Aufsichtsarbeit gestaltet werden; die Regelungen zu 1 gelten dann entsprechend.“

Die Entwicklungsarbeit ist eine Einzelarbeit.

Die Entwicklungsarbeit ist als PDF über diesen [Dateiaustauschdienst](#) spätestens bis zum **07.07.2023 (23:59 MEZ)** abzugeben.

Benennen sie ihre Datei sinnvoll (GUIPROG_EA_Name_Vorname.pdf).

Die Termine zur Demonstration ihrer Entwicklungsarbeit gebe ich rechtzeitig bekannt.

Aufgabenstellungen

Sinn und Zweck dieser Entwicklungsarbeit ist die praktische Umsetzung der in der Vorlesung behandelten Themen.

Es folgt eine Aufzählung grundsätzlicher Dinge, die sie bei der Erstellung des Codes beachten sollen. Diese Liste erhebt keinen Anspruch auf Vollständigkeit.

- Programmieren sie objektorientiert.
- Vermeiden sie redundanten Code.
- Schreiben sie modulare, überschaubare Methoden.
- Berücksichtigen sie auch das Gelernte aus dem Modul ‚Grundlagen der Informatik‘.
- Setzen sie Design by Contract ein und kommentieren sie Ihre Methoden mit Vor- und Nachbedingungen.
- Schreiben sie sinnvolle Kommentare und Dokumentationskommentare.
- Benennen sie Ihre Bezeichner sinnvoll und verständlich.
- Setzen sie Konstanten ein, da Literale im Code nur als Konstanten verwendet werden sollten. Ausnahmen sind die in der Spezifikation angegebenen Literale wie z.B. ‚null‘, ‚true‘ und ‚false‘ und Formeln mit allgemein bekannten Werten, z.B. die 2 bei $\pi/2$.
- Halten sie die Dinge einfach und verständlich. Die Performance steht nicht im Vordergrund, sondern sauberer Code.

Sie sollten sich von Quellcode, den Sie aus dem Internet kopieren, nur inspirieren lassen. Der Quellcode, den sie mir abgeben, muss an unsere Anforderungen und Maßstäbe angepasst sein.

Wir wollen in diesem Semester eine Software implementieren, mit der man Flüge einer zivilen Drohne planen, durchführen (hier nur in einer Simulation) und verwalten kann. Solch eine Drohne kann die [Mavic 3](#) von [dji](#) oder [eBeeX](#) von [AgEagle](#) sein. Zivile Drohnen können sehr nützlich sein. Zum Beispiel können sie für die Kartografierung der sich nach Vulkanausbrüchen veränderten Landschaften wie es auf La Palma geschehen ist, bei der Aufklärung der Situation nach Erdbeben oder zur Planung der optimalen Ernte in der Landwirtschaft eingesetzt werden. Die Aufgabenstellung gliedert sich in 3 Teilaufgaben.

1. Teilaufgabe: Drohnen-Controller

Sie implementieren eine Software, die die Hardware einer Drohne simuliert. Sie entscheiden, welche Hardware die Drohne haben soll. Bezüglich der Sensoren, Motoren, Aktoren, Akkus und anderer Hardware (sie können auch sehr kreativ und futuristisch sein) haben sie freie Hand.

Die Hardware der Drohne muss nicht realistisch simuliert werden. Sie werden erst in späteren Vorlesungen lernen, wie man solche Systeme modelliert und simuliert. Schauen sie sich aber diesen [Link](#) an, um mehr Einblick in die Modellierung einer Drohne zu bekommen. [Hier](#) finden sie gute Anregungen für das Model der Drohne.

Der Drohnen-Controller dient als Verbindung zwischen der Planungssoftware aus Teilaufgabe 2 und der Terrain-Simulation aus Teilaufgabe 3. Diese Netzwerkverbindungen realisieren Sie mit TCP/IP oder UDP.

2. Teilaufgabe: Planungssoftware

Implementieren Sie eine Software zur Planung, Durchführung und Verwaltung von Flügen ihrer Drohne. Sie können sich bei der Funktionalität an [eMotion](#) orientieren. Sie setzen für die GUI JavaFX ein. Ich erwarte **nicht**, dass ihre GUI professionell gestaltet ist. Jedoch sollten Sie moderne GUI-Elemente verwenden. Sie brauchen ein Menü, in dem man die einzelnen Funktionen auswählen oder Einstellungen vornehmen kann. Mit Menü meine ich ein Menü, wie es in diesem [Tutorial](#) vorgestellt wird. Sie setzen mindestens 2 Dialoge und mindestens 3 Steuerelemente davon 2 verschiedene Grundtypen (Buttons, Slider usw.) ein. Sie visualisieren das Konzept der GUI geeignet. Achten Sie darauf, dass Sie nur freie Grafiken und Sounds verwenden oder selbst welche anfertigen.

Gerne können sie sich bei der Gestaltung ihrer GUI an der [IBM-Design-Language](#), am [IBM-User-Interface-Concept](#) und an diesen [Beispielen](#) (leicht veraltet aber nützlich) orientieren.

Die anzuzeigenden Daten bekommt die Planungssoftware vom Drohnen-Controller aus Teilaufgabe 1. Die geplanten Flugdaten wie z.B. Routen und die Einstellungen der Drohne sendet die Planungssoftware an den Drohnen-Controller. Die Verbindungen sollen mit TCP/IP oder UDP realisiert werden.

3. Teilaufgabe: Terrain-Simulation

Um den Drohnen-Controller und die Planungssoftware zu testen, sollen sie einen Terrain-Simulator implementieren. Dieser soll eine Landschaft zur Verfügung stellen, in der die Drohne über Kommandos vom Drohnen-Controller bewegt werden kann und Sensordaten und Zustandsmeldungen an den Drohnen-Controller sendet. Die Kommunikation soll via TCP/IP realisiert werden.

Damit sie schnell zum Erfolg kommen, setzen sie Unity ein. Sie finden in AULIS ein Unity-Package, welches ich ihnen zusammengestellt habe. In diesem Package sind Assets, die solch einen Terrain-Simulation beispielhaft zeigen. Legen sie einfach ein neues 3D(core)-Projekt in Unity an und importieren das (Custom-)Package aus AULIS. Dazu klicken sie im Project-View mit der rechten Maustaste auf ‚Assets‘ und wählen ‚Import Package‘ aus. Öffnen sie die Szene ‚DroneExample‘ im Scenes-Ordner. In dieser Szene ist ein Terrain, eine über Tastatur steuerbare Drohne und eine Skybox. Diese Assets sind freie Assets aus dem Unity-Asset-Store. Das Skript ‚DroneServer.cs‘ stelle ich ihnen als Beispiel für eine TCP/IP Verbindung zur Verfügung. Ich wollte ihnen diese Hilfestellung geben, da in Unity C# verwendet wird. C# und Java sind zwar stark verwandt aber für den Anfang wollte ich sie mit diesem Skript unterstützen, damit sie sich schnell in Unity zurecht finden. Im Skript ‚DroneServer.cs‘ finden sie Hinweise, denen sie folgen können. Unter [Unity-Learn](#) finden sie sehr nützliche Inhalte.