

## Aufgabenstellung 2 (Gruppen B und F)

### Fachliche Aufgabenstellung

#### Einleitung

Bei der Versteigerung von Blumen werden über einen Zeitraum von mehreren Stunden Chargen, bestehend aus unterschiedlich zusammengestellten Blumengebinden, in Form einer *Rückwärtsauktion* zum Verkauf angeboten. Dabei bietet ein Auktionator ein Gebinde aus einer Charge mit einem Startpreis an, der Preis (das aktuelle Angebot) wird innerhalb einer kurzen Zeit kontinuierlich vermindert bis zu einem Mindestwert. Die Verminderung wird durch eine Art Uhr angezeigt. Der erste Kaufinteressent, der dem aktuellen Angebot zustimmt, erhält den Zuschlag. Dann erfolgt der Verkauf des nächsten Gebindes der Charge. Der Preis fällt auch mit der Anzahl der verkauften Gebinde einer Charge, dafür steigt die Mindestabnahmemenge (ursprünglich 1 Gebinde), d.h. Käufer, die lange warten, können einen niedrigeren Preis erzielen, müssen dafür aber mehr Gebinde abnehmen.

Ihre Aufgabe ist es, einen Simulator für eine solche *Rückwärtsauktion* zu erstellen. Der Simulator muss über eine ansprechende und geeignete Benutzungsschnittstelle verfügen.

#### Funktionale Anforderungen

Die Simulation soll einen vereinfachten Ablauf der Versteigerung darstellen:

- es werden einzelne Simulationsläufe durchgeführt
  - entweder laufen die verschiedenen Handlungen automatisiert ab, gesteuert durch zufällig generierte Ereignisse
    - es sollen damit 2 Käufer simuliert werden
  - oder zusätzlich mit der Möglichkeit der Benutzerinteraktion eines weiteren Käufers, d.h. es kann über die Benutzungsschnittstelle eingegriffen und ein Kauf ausgelöst werden
  - es werden N Chargen in einer Input-Queue mit unterschiedlicher Aufteilung in Gebinde eingebracht
  - bei jeder Charge werden der Anfangspreis, die Preisminderung beim Verkauf von Gebinden einer Charge (*Uhr*) und die Preisminderung von Verkauf zu Verkauf der Gebinde einer Charge festgelegt
- in einem Simulationslauf werden alle Gebinde und alle Chargen verkauft, ggf. werden mehrere Auktionsrunden für eine Charge durchgeführt
- die Durchführung der Simulation wird dokumentiert; dazu wird ein Ablaufprotokoll in einer Textdatei gespeichert.

Die Beschreibung der Chargen wird serverseitig in JSON-formatierten Dateien abgelegt und zu Beginn des Simulationslaufs eingelesen. Diese Daten zur Beschreibung der Chargen und alle anderen Daten werden interaktiv bearbeitet.

#### Anforderungen an die Gestaltung der Benutzungsschnittstelle

##### Allgemeine Anforderungen

Die Benutzungsschnittstelle muss die Konfiguration der Simulationen und die Darstellung der Durchführung wirkungsvoll unterstützen. Der Aspekt Barrierefreiheit muss, soweit das fachlich möglich ist, beachtet werden.

Beachten Sie die Anwendung der Gestaltgesetze.

##### Umfang

Zur Benutzungsschnittstelle gehören:

- die Pflege der Daten zu den einzelnen Chargen, die in der Input-Queue für die Simulationen verwendet werden

- als Listensicht
- als Detailsicht
- die Bearbeitung der Simulationsparameter in einem Formular
- die Darstellung des Simulationsablaufs einschließlich der Möglichkeit des manuellen Eingriffs
- ein Anwendungsrahmen, von dem aus die anderen Sichten erreichbar sind und eine Simulation gestartet werden kann
- die Anzeige des Ablaufprotokolls mit der Möglichkeit, die Datei zur Speicherung der Ablaufprotokolls auszuwählen.

Zur Darstellung der Preisänderung müssen Sie eine analoge und eine digitale Anzeige vorsehen:

- eine *Uhr*:
  - Startpunkt ist die Position "12 Uhr"
  - die Anzeige läuft gegen den Uhrzeigersinn
  - jede "Sekunde" entspricht einer Preisänderung entsprechend den eingestellten Vorgaben
- eine digitale Anzeige des aktuellen Preises, die der Position der Uhr entspricht.

Ferner müssen folgende Angaben bei der Simulation angezeigt werden:

- Angaben zur Charge / Gebinde des aktuellen Verkaufs
- die noch in der Input-Queue vorhandenen Chargen
- die bereits verkauften Chargen / Gebinde, aufgeteilt nach Käufer sowie summarische Angaben (Anzahlen, Summe erzielter Preis).

## Anforderungen an die Umsetzung

Die Anwendung wird als Web-Anwendung realisiert:

- verwenden Sie HTML5
- verwenden Sie CSS (siehe auch Hinweis beim Abschnitt *Hilfsmittel*)
- verwenden Sie zum Datenaustausch mit dem Web-Server Ajax / JSON
- Daten werden auf dem Web-Server im JSON-Format gespeichert; das Ablaufprotokoll kann als reine Textdatei gespeichert werden
- zur Darstellung der Uhr verwenden Sie die Möglichkeiten des canvas-Elements (siehe auch Hinweis beim Abschnitt *Hilfsmittel*).

Zur Implementierung zeitabhängiger Ereignisse nutzen Sie die DOM-Methoden `setTimeout` und `setInterval`. Es ist empfehlenswert, sich mit dem Konzept *Web-Worker* auseinanderzusetzen. Damit besteht die Möglichkeit, parallele Verarbeitungsvorgänge zu implementieren (siehe z.B. [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Workers\\_API/Using\\_web\\_workers](https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers) und als kurze deutschsprachige Erläuterung <http://www.mediaevent.de/javascript/web-worker.html>).

## Anforderungen an die Durchführung

### Termin A

Bei diesem Termin legen Sie Ihren Entwurf vor. Geben Sie dazu an:

- ein Zustandsmodell der Anwendung
- ein wireframe-Entwurf der Anwendung (Anwendungsrahmen, Formulare/Listen, Darstellung Simulationsablauf)
- eine Spezifikation der notwendigen Datenstrukturen und Daten
- einen Prototypen für den Kern der Simulation, insbesondere die Simulation der Läufer und die Steuerung der Simulation.

### Termin B

Bei diesem Termin

- stellen Sie Ihre Simulationsanwendung vor und erläutern diese
- legen Sie Ihre Dokumentation vor und erläutern diese.

## Anforderungen an die Dokumentation

### Allgemeine Anforderungen

Verwenden Sie als *Ausgangspunkt* für die Strukturierung Ihrer Dokumentation folgende Gliederung:

1. Einleitung
2. Entwurf der Anwendung
  - 2.1 Benutzungsschnittstelle
    - 2.1.1 Anwendungsrahmen und einzelne Sichten
    - 2.1.2 Zustandsmodell
  - 2.2 Steuerung der Simulation
  - 2.3 Datenaustausch Client - Server
3. Implementierung der Anwendung
  - 3.1. Web-Client
    - 3.1.1 Benutzungsschnittstelle
    - 3.1.2 Datenaustausch mit dem Server
  - 3.2. Web-Server
    - 3.2.1 Bearbeitung der Anfragen
    - 3.2.2 Speicherung von Daten
4. Durchführung von Simulationen

Die Dokumentation wird als utf-8 kodierter Text mit der einfachen Auszeichnungssprache *markdown* erstellt. Mit Hilfe des Werkzeugs *pandoc* (siehe Hilfsmittel) erfolgt die Umsetzung in eine HTML-Datei:

```
pandoc -f markdown -t html5 -s -c iasp1.css --toc <IhreDatei> -o <IhreHTML5Datei>
```

Die Datei *iasp1.css* enthält zusätzliche CSS-Stilregeln und wurde Ihnen bereits mit der Aufgabe 1 zur Verfügung gestellt.

Beachten Sie folgende Anforderungen:

- das Werkzeug *pandoc* **muss** genutzt werden
- **die Erstellung der Dokumentation mit anderen *markdown*-Werkzeugen / Editoren wird nicht zugelassen**
- **die in *pandoc* verfügbaren Erweiterungen der Auszeichnungssprache *markdown* müssen genutzt werden**, z.B. bei der Verwendung von Tabellen
- die Zustandsdiagramme, die Sie mit *UMLet* erstellen, binden Sie als Graphiken ein
- exportieren Sie dazu die Diagramme als PNG-Dateien.

### Zustandsmodell

Erstellen Sie das Zustandsmodell mit Hilfe von UML-Diagrammen des Typs *UML-State-Machine*. Wählen Sie bei *UMLet* in der Auswahlliste (rechts oben) diesen Diagramm-Typ aus.

Jede Sicht / Webseite wird als *Zustand* modelliert. Die Zustandsübergänge repräsentieren die vorgesehenen Bedienungen z.B. von Verweisen oder Schaltern.

Eine ausführliche Beschreibung finden Sie in den Materialien zum Buch "UML@Classroom" (siehe <http://www.uml.ac.at/de/lernen>, dort: "Zustandsdiagramm").

## Hilfsmittel

Sie **müssen** folgende Hilfsmittel verwenden:

### Erstellung Zustandsmodell

Erstellen Sie Zustandsdiagramme mit dem Werkzeug *UMLet* (siehe <http://www.umlet.com>).

### Erstellung Dokumentation

Verwenden Sie *pandoc* (siehe <http://pandoc.org/>) zur Konvertierung der Markdown-Datei in eine HTML5-Datei.

### Implementierung

Verwenden Sie:

- serverseitig:
  - Python
  - das Python-Framework *cherry*
- clientseitig:
  - javascript
  - jquery
  - AJAX
  - JSON als Format zum Datenaustausch
  - oCanvas (siehe <http://ocanvas.org>) zum Zeichnen im canvas-Element
  - inheritance.js, es.js (wird Ihnen mit dieser Aufgabenstellung bereitgestellt)
  - uikit (CSS-Framework, siehe <http://getuikit.com>)

## Testat

Um das Testat zu erhalten, müssen Sie

- beide Termine wahrnehmen
- beim ersten Termin Ihren Entwurf vorlegen und erläutern
- beim zweiten Termin
  - Ihre Simulationsanwendung vorlegen, vorführen und erläutern
  - Ihre Dokumentation vollständig vorlegen und erläutern.