

Pflichtenheft

**Augmented Virtual
Reality**

V1.0

- I. **Zielbestimmung**
 - a. Musskriterien
 - b. Wunschkriterien
 - c. Abgrenzungskriterien
- II. **Produkteinsatz**
 - a. Anwendungsbereiche
 - b. Zielgruppen
 - c. Betriebsbedingungen
- III. **Produktumgebung**
 - a. Software
 - b. Hardware
 - c. Produkt-Schnittstellen
- IV. **Funktionale Anforderungen**
- V. **Produktdaten**
- VI. **Nichtfunktionale Anforderungen**
- VII. **Systemmodelle**
 - a. Szenarien
 - b. Anwendungsfälle
 - c. Benutzerschnittstelle – Bildschirmskizzen, Navigationspfade
- VIII. **Globale Testfälle**
- IX. **Entwicklungsumgebung & zusätzl. Tools**
- X. **Glossar**

Einleitung

Im Rahmen der Veranstaltung PSE ist es unsere Aufgabe, eine Software zu entwickeln. Die Software ermöglicht den Mitarbeitern bzw. Gästen von TecO, sich die Sensordaten in den Büros über eine GUI mit Hilfe einer virtuellen Umgebung anzeigen zu lassen. Die virtuelle Umgebung bildet dabei die realen Räume nach und wertet die Darstellung mit einer visuellen Interpretierung der Sensordaten auf.

So soll zum Beispiel erkennbar sein, welche Raumtemperatur gerade erfasst wird oder ob das Licht eingeschaltet ist.

1. Zielbestimmung

1.a Musskriterien

- eine Übersichtsanzeige des Grundrisses
- laden des Grundrisses von TecO
- eine Visualisierung der Sensordaten
- Erweiterbarkeit durch Plugins
- 3D Darstellung der Räumlichkeiten
- Freie Bewegung in der virtuellen Umgebung muss möglich sein
- verschiedene Sensortypen müssen integrierbar sein (über Plugins)
- Auswahlmöglichkeit von einzelnen Sensortypen

1.b Wunschkriterien

- Ansteuerung von Aktuatoren über die GUI
- Transparente Wände (auf Knopfdruck)
- Zeitlicher Verlauf von Sensordaten
- Möglichkeit alternative Grundrisse anzuzeigen
- Defekte Sensoren anzeigen
- Anzeigen von belegten Räumen mit Hilfe der Sensoren
- Zugriff über ein Webinterface
- Verschiebung von Sensoren/Objekten

1.c Abgrenzungskriterien

- Raumbelegung der Seminarräume über GUI (muss separat eingebunden werden)
- Sensoren werden nicht selber eingetragen (werden über Schnittstelle übergeben)

2. Produkteinsatz

2.a Anwendungsbereiche

Das Produkt wird eingesetzt um Sensordaten mittels GUI in einer virtuellen Umgebung anzuzeigen und optional um die Aktuatoren zu steuern. Die Büroräume von TecO mit den Sensoren werden virtuell dargestellt. Die Sensordaten werden in unterschiedlichen Darstellungsformen angezeigt, welche von Plugins gesondert festgelegt werden können - so ist das System leicht erweiterbar und neue Sensortypen können leicht integriert werden. Optional ist das System über ein Webinterface abrufbar.

2.b Zielgruppen

- Mitarbeiter von TecO (Vollzugriff)
- Gäste von TecO (Nur Zugriff auf die Daten. Keine Änderungen möglich)

2.c Betriebsbedingungen

Das Hauptziel ist der Einsatz in den Büroräumen von TecO, optional soll aber auch der Einsatz in beliebigen Räumen, die entsprechend ausgestattet sind, möglich sein. Es muss ein Zugriff auf Sensordaten über das SmartTecO System möglich sein, über das die Software die Sensordaten beziehen kann.

3. Produktumgebung

Das Programm soll auf einem Rechner mit Javaumgebung laufen.

3.a Software

- Betriebssystem mit Java Laufzeitumgebung in der Mindestversion 6
- OpenGL in der Mindestversion 3.0

3.b Hardware

- PC mit Mindestanforderungen (siehe 6.)

3.c Produkt-Schnittstellen

- Plugin Schnittstelle
Die Plugins ermöglichen die Darstellung verschiedener Sensortypen.
- Anbindung an ein SmartTecO System zur Anbindung der Sensoren.
Diese werden über die REST Schnittstelle abgefragt.
- Webinterface für optionale Webanbindung
Ein kleiner Webserver macht die Sensordaten von außen zugänglich.

4. Funktionale Anforderungen

Musskriterien des Grundsystems:

- /F010/ Laden der Umgebung aus einer Eingabedatei
Das Programm erzeugt aus einer XML-Eingabedatei eine virtuelle Umgebung. In der XML Datei sind Wände, Fenster und Raumhöhe, sowie Referenzpunkte in globalen lat/lon Koordinaten spezifiziert, sodass die Sensoren richtig eingeordnet werden können.
- /F020/ Grafische 3D-Darstellung der Umgebung
Die erzeugte Umgebung wird angezeigt und eine Benutzeroberfläche eingeblendet. Das Bild soll in Echtzeit aktualisiert werden.
- /F030/ Übersicht ein- und ausblenden
Der Benutzer soll die Möglichkeit besitzen, zwischen Ego- und Vogelperspektive zu wechseln. In der Vogelperspektive werden Decken ausgeblendet, sodass der Nutzer möglichst freie Sicht auf alle Räume und Sensoren hat.
- /F040/ Bewegung in der virtuellen Umgebung
Der Benutzer soll sich in der virtuellen Umgebung per Tastatur/Maus aus der Egoperspektive frei bewegen können. Dabei werden aber Kollisionen berechnet, sodass er nicht durch Wände oder Gegenstände hindurchlaufen kann. Auch Schwerkraft wird miteinberechnet, um eine möglichst realitätsgetreue Umgebung zu schaffen.
- /F050/ Bereitstellung von Schnittstellen für Erweiterbarkeit
Das Programm soll in der Lage sein externe Plugins zur Erweiterung der Funktionalität zu laden und auszuführen. Das geladene Plugin ist für die Darstellung von Sensordaten für einen bestimmten Sensortyp zuständig.
- /F060/ Visualisierung der Sensordaten über Plugins
Die Plugins übernehmen die Darstellung von Sensoren mit definierbaren Tags und zeigen diese in der Umgebung an. Jedes Plugin kann entscheiden, auf welche Tags es zugreifen will. Pro Sensortyp soll ein eigenes Plugin die Darstellung übernehmen.
- /F070/ Laden der Sensordaten
Die Sensordaten werden über die TecO-API abgefragt und als JSON Stream verarbeitet. Aus dem JSON Stream werden die Sensoreigenschaften ausgelesen und intern zwischengespeichert.
- /F080/ Speicherung von früheren Sensorwerten in einer Datenbank
Sensordaten werden protokolliert und in eine Datenbank für späteren Gebrauch gespeichert. Diese werden den Plugins zugänglich gemacht, damit diese statistische Diagramme anzeigen können.

Wunschkriterien des Grundsystems:

- /F090/ Steuerung von Aktuatoren
Der Benutzer kann vorhandene Aktuatoren über das GUI steuern.
- /F100/ Speicherung der Benutzereinstellungen
Benutzereinstellungen werden in einer XML-Datei gespeichert und beim Start geladen.
- /F110/ Unterstützung von Shortcuts
Der Benutzer soll per Tastaturkürzel auf bestimmte Funktionen des Grundsystems zugreifen können.

- /F120/ Verschiebung von Sensoren
Der Benutzer soll in der Lage sein Sensoren zu verschieben.
- /F130/ Wände transparent machen
Der Benutzer soll durch einen Knopfdruck oder ein Shortcut die Wände transparent gestalten können, um Sensoren, die hinter einer Wand liegen, sichtbar zu machen.
- /F140/ Defekte Sensoren anzeigen
Sensoren, die keine richtigen oder gar keine Daten liefern, sollen rot blinkend als defekt markiert werden. Jedes Plugin kann eigene Richtlinien bestimmen, nach welchen es einen Sensor als defekt einstuft.
- /F150/ Zugriff auf GUI über Webinterface
Der Benutzer soll in der Lage sein über einen Browser das Programm aufzurufen.

Funktionale Anforderungen des Plugins:

Musskriterien:

- /F160/ Die Temperatur wird mit Farben dargestellt (Heizung)
Je wärmer die Heizung wird, desto röter wird die Heizung angezeigt. Je kälter, desto blauer wird die Heizung. blau($\leq 5^{\circ}\text{C}$) rot($\geq 40^{\circ}\text{C}$)

Wunschskriterien:

- /F170/ Positionsveränderung der Heizung
Die Heizung soll in der GUI mit der Maus verschoben werden, die Positionsdaten zurückliefern und in den Sensordaten gespeichert werden.

5. Produktdaten

Produktdaten des Grundsystems:

Diese Daten sind zwingend notwendig, es sei denn sie sind als optional gekennzeichnet.

- /D010/ Sensordaten
Es liegen die Daten der Sensoren vor und können benutzt werden.
 - Typ
 - Tag
 - ID
 - Ort
 - Beschreibung (optional)
 Alle Sensordaten werden für eine definierbare Zeit in einer Datenbank hinterlegt.
- /D020/ Benutzereinstellungen (Optional)
Der Benutzer kann auf seine vorherigen Einstellungen zurückgreifen.
 - Dokumententyp (XML)
 - Administrator (optional)
 - Dokumentengröße (optional)
 Optional werden Benutzereinstellungen wieder als XML-Datei gespeichert.

- /D030/ Grundriss
Es liegt ein Grundriss vor, der von der Software gelesen werden kann.
 - Grundriss-XML
 - Dokumentgröße (optional)
 - Beschreibung (optional)

Produktdaten des Plugins:

- /D040/ Heizungsmodell
Das 3D-Modell mit Textur der Heizung ist gespeichert.

6. Nichtfunktionale Anforderungen

- /NF10/ Die Darstellung der Umgebung soll auf einem Rechner mit Standardhardware in Echtzeit erfolgen und mindestens 30 Bilder pro Sekunde erreichen. Dabei sollen folgende Mindestanforderungen gelten:
 - Dualcore Prozessor mit mindestens 2 GHz
 - OpenGL 3.0 fähiger Grafikprozessor
 - 1 GB RAM
- /NF20/ Das Programm muss unter Windows, Linux und Mac OS X laufen.

7. Systemmodelle

a) Szenarien:

1.) Eine Person befindet sich auf dem Heimweg. Plötzlich fällt ihr ein dass sie möglicherweise das Licht im Arbeitszimmer angelassen hat. Da sie sich nicht sicher ist, holt sie ihren Laptop raus, startet das Programm. Sie lädt den Grundriss über eine XML ein, klickt auf die Übersicht und schaut sich erstmal alles an. Dann schließt sie die Übersichtsseite und stellt fest, dass das Licht an ist. Also schaltet sie es aus und testet noch einige Tastenkürzel, bevor sie das Programm schließt.

2.) Eine Person liegt krank im Bett. Ihr ist kalt. Sie möchte die Heizung aufdrehen. Sie startet ihren Laptop und das Programm. Sie stellt die Heizung auf 25°. Dann entscheidet sie sich noch die Daten des Nachmittags als Diagramm nachzusehen und beendet das Programm.

3.) Eine Person startet das Programm. Sie bewegt sich durch die virtuelle Welt und sieht, dass jemand im Wohnzimmer sitzt. Sie erlaubt sich einen Scherz und schaltet das Licht aus. Sie läuft virtuell im Raum umher gegen jede Wand und das ganze Inventar.

b) Anwendungsfälle:

1. dreidimensionale Darstellung der Umgebung

Ziel: Räume aus .xml Datei in eine 3D Umgebung umwandeln

Kategorie: Musskriterium

Vorbedingungen: .xml Datei mit Grundriss vorhanden

Nachbedingung (Erfolg): GUI stellt Gebäude dreidimensional dar

Nachbedingung (Fehlschlag): Fehlermeldung wird angezeigt

Akteure: Benutzer

Auslösendes Ereignis: starten der GUI

Beschreibung:

Der Benutzer startet die GUI und lädt die .xml des Grundrisses.

2. Plugin aktivieren

Ziel: Plugin aktivieren

Kategorie: Musskriterium

Vorbedingungen: Plugins vorhanden und hinzugefügt.

Nachbedingung (Erfolg): Plugin wird geladen

Nachbedingung (Fehlschlag): Fehlermeldung wird angezeigt

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer wählt aus der Pluginliste das Plugin aus, das er aktivieren möchte.

3. Plugin deaktivieren

Ziel: Plugin deaktivieren

Kategorie: Musskriterium

Vorbedingungen: Plugins vorhanden und aktiviert.

Nachbedingung (Erfolg): Plugin wird beendet

Nachbedingung (Fehlschlag): Fehlermeldung wird angezeigt

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer wählt aus der Pluginliste das Plugin aus, das er deaktivieren möchte.

4. Bewegen in der virtuellen Umgebung

Ziel: Freies bewegen in der virtuellen Umgebung

Kategorie: Musskriterium

Vorbedingungen: angeschlossene Tastatur und Maus

Nachbedingung (Erfolg): von einem Punkt in der Umgebung zu einem anderen bewegt

Nachbedingung (Fehlschlag): nicht bewegt

Akteure: Benutzer

Auslösendes Ereignis: Eingabe über die Tastatur

Beschreibung:

Der Benutzer drückt eine der für die Bewegung definierten Tasten und bewegt sich durch die virtuelle Umgebung.

5. Alte Sensordaten anzeigen

Ziel: Zugriff auf Sensordaten in der Vergangenheit

Kategorie: Musskriterium

Vorbedingungen: Sensor vorhanden

Nachbedingung (Erfolg): Der Benutzer sieht einen Graphen mit allen Sensordaten der Vergangenheit

Nachbedingung (Fehlschlag): -

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer klickt einen Sensor an und wählt die Statistikfunktion aus.

6. Übersicht einblenden

Ziel: Übersicht des Grundrisses einblenden

Kategorie: Musskriterium

Vorbedingungen: keine

Nachbedingung (Erfolg): Der Benutzer sieht eine Übersicht aus Vogelperspektive der gesamten Umgebung

Nachbedingung (Fehlschlag): Der Benutzer sieht keine Übersicht.

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer drückt auf den Übersichtsbutton im GUI um eine Übersicht über alle Sensoren zu bekommen.

7. Raum auswählen

Ziel: Sprung in einen beliebigen Raum

Kategorie: Wunschkriterium

Vorbedingungen: Der Benutzer befindet sich in der Übersichtsansicht

Nachbedingung (Erfolg): Der Benutzer befindet sich im ausgewählten Raum

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer klickt einen Raum an.

8. Aktuator steuern

Ziel: Geräte in der realen Umgebung ansteuern

Kategorie: Wunschkriterium

Vorbedingungen: ein entsprechender Aktuator ist vorhanden

Nachbedingung (Erfolg): Der Aktuator hat seinen Zustand geändert

Nachbedingung (Fehlschlag): Der Aktuator hat seinen Zustand nicht geändert

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer drückt auf ein Gerät in der virtuellen Umgebung und wählt eine Aktuatorfunktion an um den Zustand des Aktuators zu ändern.

9. Sensorposition verändern

Ziel: Verschieben eines Sensors

Kategorie: Wunschkriterium

Vorbedingungen: -

Nachbedingung (Erfolg): Der Sensor befindet sich an einer neuen Position

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer verschiebt einen Sensor weil sich die Position des Sensors in der echten Welt geändert hat.

10. Einstellungen des Benutzers werden gespeichert

Ziel: Erstellen einer .xml Datei mit den Einstellungen des Benutzers

Kategorie: Wunschkriterium

Vorbedingungen: GUI muss gestartet sein

Nachbedingung (Erfolg): Eine .xml mit den Benutzerdaten wurde angelegt.

Nachbedingung (Fehlschlag): Fehlermeldung wird angezeigt

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer geht über das Menü in Einstellung speichern und speichert damit alle Einstellungen.

11. Shortcut benutzen

Ziel: Shortcuts sollen nutzbar sein

Kategorie: Wunschkriterium

Vorbedingungen: -

Nachbedingung (Erfolg): Shortcut wird erfolgreich ausgeführt.

Nachbedingung (Fehlschlag): es passiert nichts

Akteure: Benutzer

Auslösendes Ereignis: drücken des Shortcuts

Beschreibung:

Der Benutzer drückt ein beliebiges Shortcut mit einer Funktion und führt damit eine Aktion durch.

12. Sensor verschieben

Ziel: Man soll in der Lage sein Sensorpositionen über die GUI zu ändern

Kategorie: Wunschkriterium

Vorbedingungen: -

Nachbedingung (Erfolg): Positionsdaten wurden in der JSON Datei geändert.

Nachbedingung (Fehlschlag): Positionsdaten sind in der JSON Datei unverändert.

Akteure: Benutzer

Auslösendes Ereignis: Drag&Drop

Beschreibung:

Der Benutzer zieht per Drag&Drop-Verfahren den Sensor an die gewünschte Position und die neue Position wird an die SensorAPI zurück geliefert und dort gespeichert.

13. Wände transparent machen

Ziel: Die Wände sollen transparent schaltbar sein um dahinter liegende Sensoren sichtbar zu machen.

Kategorie: Wunschkriterium

Vorbedingungen: -

Nachbedingung (Erfolg): Die Wände werden auf Kopfdruck transparent.

Nachbedingung (Fehlschlag): Fehlermeldung wird angezeigt

Akteure: Benutzer

Auslösendes Ereignis: Klick

Beschreibung:

Der Benutzer in einem Menü "Wände transparent machen", woraufhin die Wände transparent werden.

14. Defekte Sensoren anzeigen

Ziel: Defekte Sensoren werden rotblinkend angezeigt

Kategorie: Wunschkriterium

Vorbedingungen: -

Nachbedingung (Erfolg): ein defekter Sensor wird rot blinkend dargestellt

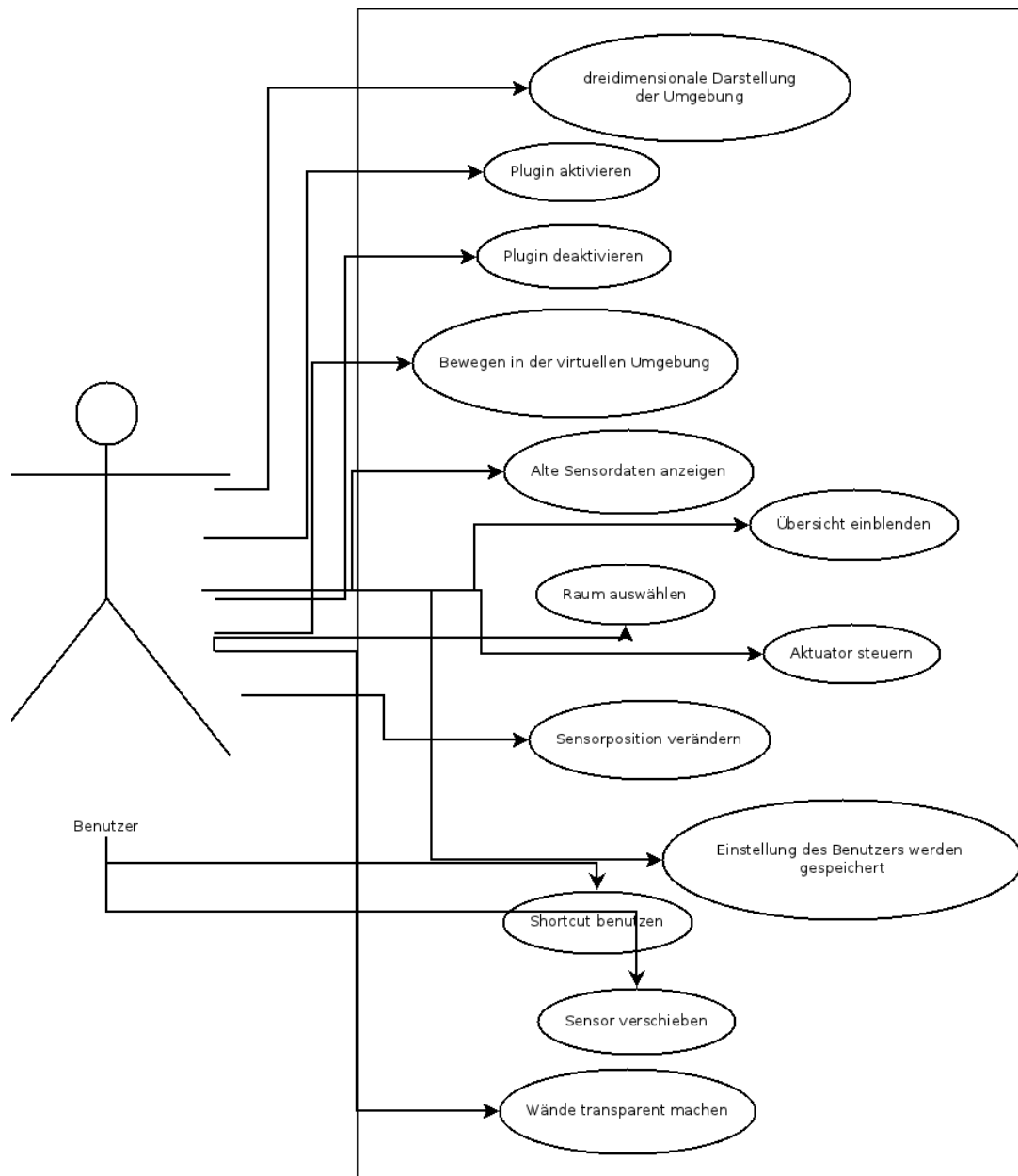
Nachbedingung (Fehlschlag): es passiert nichts

Akteure: Grundsystem

Auslösendes Ereignis: falsche oder gar keine Daten werden dem Plugin übergeben

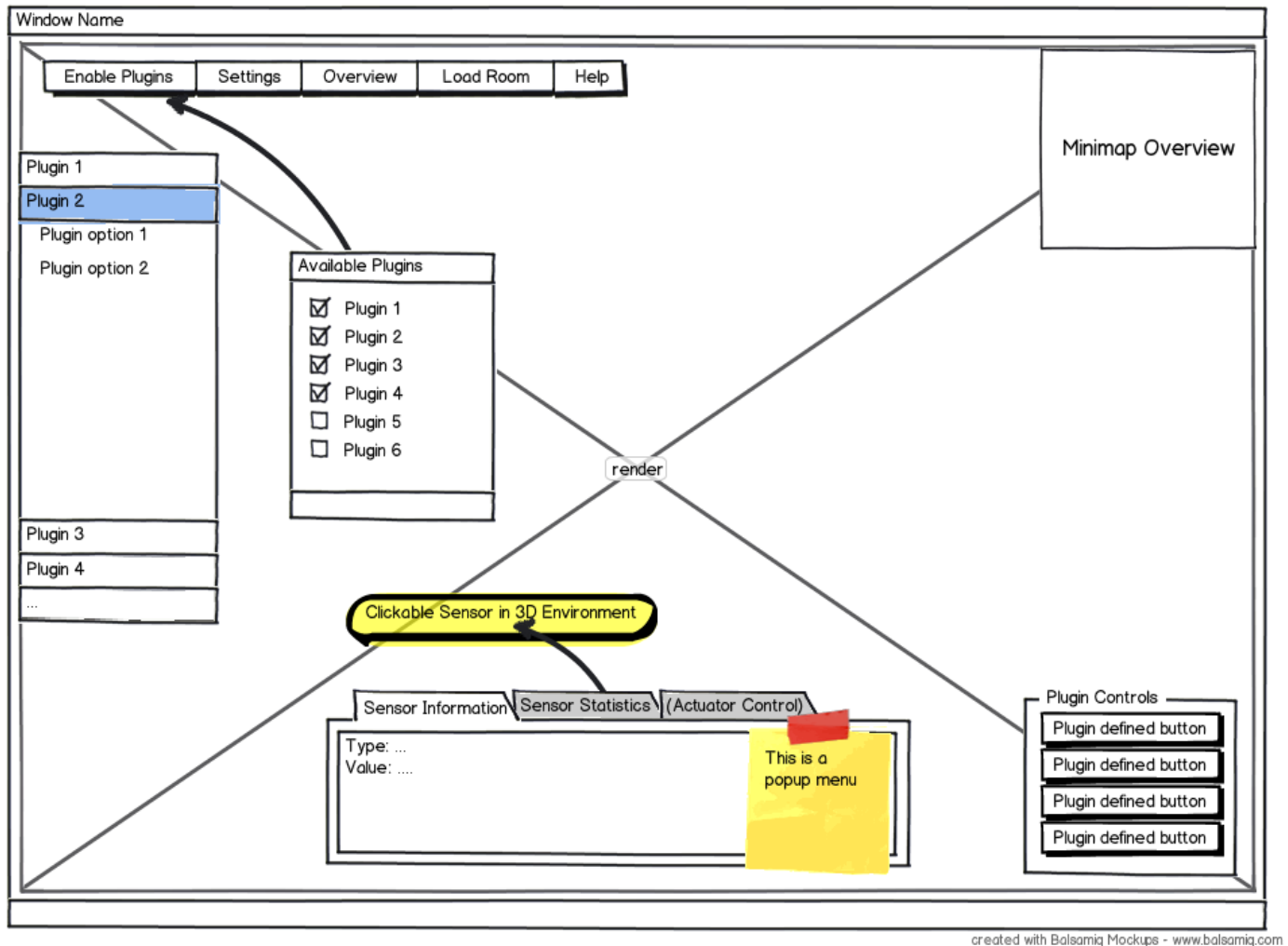
Beschreibung:

Wenn ein Sensor defekt ist, sendet er unkorrekte oder gar keine Daten mehr. Daraufhin wird in der GUI der Sensor rotblinkend angezeigt.



c) Benutzerschnittstelle

Der Benutzer interagiert mit der Software mittels einer Menüstruktur, welche direkt in der grafischen Oberfläche eingeblendet wird.

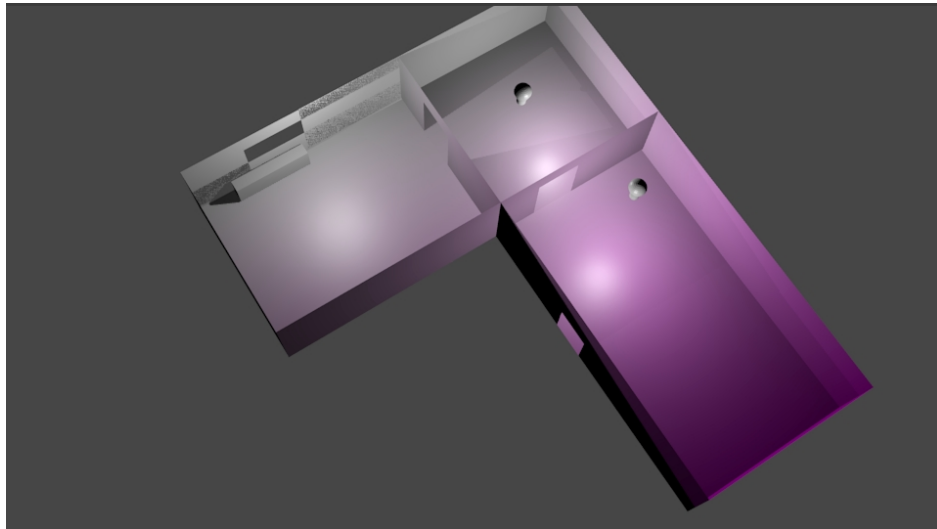


Über den Button “Enable Plugins” gelangt der Benutzer zu einem Dialog, der ihm erlaubt vorhandene Plugins zu aktivieren oder zu deaktivieren. Aktivierte Plugins können auf der linken Bildschirmseite ein Optionsmenü bereitstellen um bestimmte Parameter anzupassen und in der rechten unteren Bildschirmecke Buttons für Aktionen anzeigen.

In der oberen Menüleiste befinden sich auch die anderen Funktionen des Grundsystems:

- “Settings” ruft ein Einstellungsmenü auf, über welches der Benutzer zum Beispiel die Bildschirmauflösung einstellen kann.
- “Overview” wechselt zwischen Vogel- und Egoperspektive
- “Load Room” ist für das laden eines Grundrisses zuständig, den der Benutzer auswählen kann.
- über “Help” gelangt der Nutzer zu einer kurzen Programmeinführung, die das Bedienkonzept erklärt

Oben rechts wird eine “Minimap” angezeigt, die eine Übersicht der Räume darstellt.



Sobald der Nutzer einen Sensor auswählt, erscheint ein Kontextmenü mit pluginspezifischen Inhalten, die genauere Informationen über den Messwert enthalten.

In der Egoperspektive kann der Benutzer sich per Tastatur frei in der Umgebung bewegen und mit der Maus umschauen.

d) Architektur der Software:

Die Software besteht aus dem Grundsystem und den Plugins. In dem Grundsystem werden die Daten der Sensoren zwischengespeichert. Außerdem ist das Grundsystem dafür verantwortlich aus der eingelesenen .xml Datei des Grundrisses eine dreidimensionale Umgebung zu generieren. Die Plugins werden vom Nutzer in ein dafür vorgesehenes Verzeichnis eingefügt und beim Start der Software automatisch von dort aus geladen. Diese können dann vom Benutzer nach Wahl angezeigt oder ausgeblendet werden. Die Plugins können selbst entscheiden, welche Tags und Daten sie verarbeiten wollen. Hierfür registrieren sich die Plugins nach dem Beobachtermodell beim Grundsystem für bestimmte Tags, mit denen Sensoren ausgestattet sind. Sobald sich der Zustand eines Tags ändert, werden alle hierfür registrierten Plugins benachrichtigt.

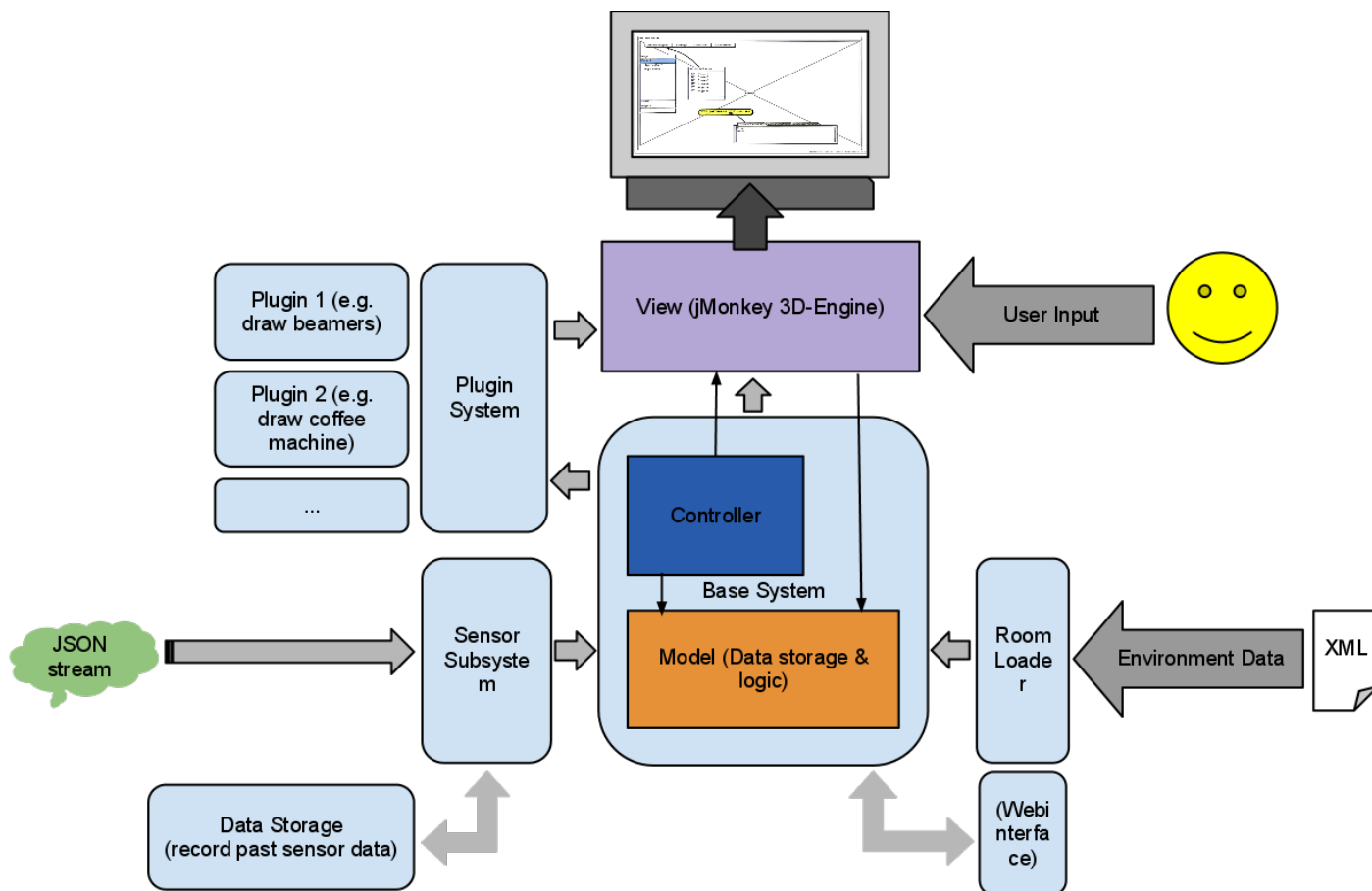
Bei jedem Rendervorgang werden alle aktivierten Plugins der Reihe nach aufgerufen, um die von ihnen verwalteten Sensordaten anzuzeigen.

Die Plugins sind dann selbstständig dafür verantwortlich, die Visualisierungen an der richtigen Stelle in der 3D-Umgebung zu zeichnen. Somit haben Grundsystem und Plugin Zugriff auf die Render Engine und können getrennt voneinander in die 3D-Umgebung zeichnen. Das Grundsystem ist auch für das Anlegen einer Datenbank

verantwortlich, in der alle bisher empfangenen Daten gespeichert werden. Wenn der Benutzer einen Graphen über den Verlauf anzeigen will, greift das für den Sensor verantwortliche Plugin auf das Grundsystem zu, um die Sensordaten anzufordern, und zeichnet den Graphen in die GUI.

Das Grundsystem setzt sich nach dem Model-View-Controller Entwurfsmuster zusammen und ist wie folgt aufgetrennt:

- Das von der Render Engine verwaltete GUI repräsentiert das View
- Das Model enthält den Zwischenspeicher aller verfügbaren Sensordaten
- Der Controller koordiniert das Model und das View



8. Globale Testfälle

Testfälle des Grundsystems :

Musskriterien:

- /T010/ Umgebung aus einer Eingabedatei laden
- /T020/ Anzeige überprüfen
- /T030/ Übersicht einblenden/ausblenden
- /T040/ Läuft durch die Räume, gegen Wände und Inventar
- /T050/ Plugins an- und ausschalten
- /T060/ Sensordaten darstellen lassen über Plugins
- /T070/ Aktuelle Sensordaten abfragen
- /T080/ Alte Sensordaten abfragen
- /T090/ Programm einschalten/ausschalten
- /T100/ Diagrammdarstellung ausprobieren

Wunschkriterien:

- /T110/ Aktuatoren steuern
- /T120/ Speichern der Einstellungen
- /T130/ Tastaturkürzel benutzen
- /T140/ Sensoren verschieben
- /T150/ Knopf drücken und Wand transparent machen
- /T160/ schauen ob Sensor rot blinkt
- /T170/ über das Internet auf die Daten zugreifen

Testfälle des Plugins:

Musskriterien:

- /T180/ Die Temperatur abfragen und dabei die Farben beachten.

Wunschkriterien:

- /T190/ Die Position der Heizung verschieben.

9. Entwicklungsumgebung & zusätzliche Tools

- 1. Entwicklungssoftware

- Eclipse

- Verwendete Plugins:

- Checkstyle
 - Findbugs
 - PMD
 - EMMA

- JUnit

- Für Unit tests wird das Framework JUnit verwendet, welches in Eclipse integrierbar ist.

-EMMA

Für Überdeckungstests setzen wir EMMA ein.

- Javadoc

Zur Dokumentation des Codes kommen klassische Javadoc
kommentare zum Einsatz.

- 2. Versionsverwaltung

Zur Versionsverwaltung wird das Tool “git” benutzt

10. Glossar

- GUI
GUI (=Graphical User Interface): Grafische Benutzeroberfläche
- Plugin:
Programmerweiterung
- Aktuator:
Steuerung von physikalisch realen Objekten,
- Webinterface:
Schnittstelle für das Internet
- API:
Application Programming Interface: Schnittstelle für Sensordaten
- JSON Stream
JSON (=JavaScript Object Notation): Übertragungsformat für Daten.
- Shortcut
Tastaturkürzel
- Orgware:
Beschreibt die Rahmenbedingungen der Software.
- Minimap
2-dimensionale Übersichtskarte
- Render Engine
Bibliothek für grafische Darstellungen.