



Projekt: JBomberman  
Software Architektur

Pascal Kistler  
Silvan Adrian  
Fabian Binna

## 1 Änderungshistorie

Datum	Version	Änderung	Autor
01.04.15	1.00	Erstellung des Dokuments	Gruppe
05.04.15	1.01	Logische Architektur	Fabian Binna

## Inhaltsverzeichnis

<b>1</b>	<b>Änderungshistorie</b>	<b>2</b>
<b>2</b>	<b>Einführung</b>	<b>4</b>
2.1	Zweck . . . . .	4
2.2	Gültigkeitsbereich . . . . .	4
2.3	Referenzen . . . . .	4
2.4	Übersicht . . . . .	4
<b>3</b>	<b>Systemübersicht</b>	<b>4</b>
<b>4</b>	<b>Architektonische Ziele &amp; Einschränkungen</b>	<b>4</b>
<b>5</b>	<b>Logische Architektur</b>	<b>5</b>
5.1	<Presentation/view> . . . . .	6
5.1.1	Klassenstruktur . . . . .	6
5.1.2	Schnittstellen . . . . .	7
5.1.3	Wichtige interne Abläufe . . . . .	7
5.2	Wichtige Abläufe . . . . .	7
<b>6</b>	<b>Prozesse und Threads</b>	<b>7</b>
<b>7</b>	<b>Deployment</b>	<b>7</b>
<b>8</b>	<b>Datenspeicherung</b>	<b>7</b>
<b>9</b>	<b>Größen und Leistung</b>	<b>7</b>

## 2 Einführung

### 2.1 Zweck

Dieses Dokument beschreibt die Software Architektur für das Projekt JBombberman.

### 2.2 Gültigkeitsbereich

Dieses Dokument ist während des ganzen Projekts gültig und wird laufend aktualisiert.

### 2.3 Referenzen

<Liste aller verwendeten und referenzierten Dokumente, Bücher, Links, usw.> <Referenz auf ein Glossar Dokument, wo alle Abkürzungen und unklaren Begriffe erklärt werden>  
<Die Quellen / Referenzen sollten mit dem Word Tool automatisch erstellt werden>

### 2.4 Übersicht

<Übersicht über den restlichen Teil dieses Dokumentes geben und dessen Aufbau erläutern>

## 3 Systemübersicht

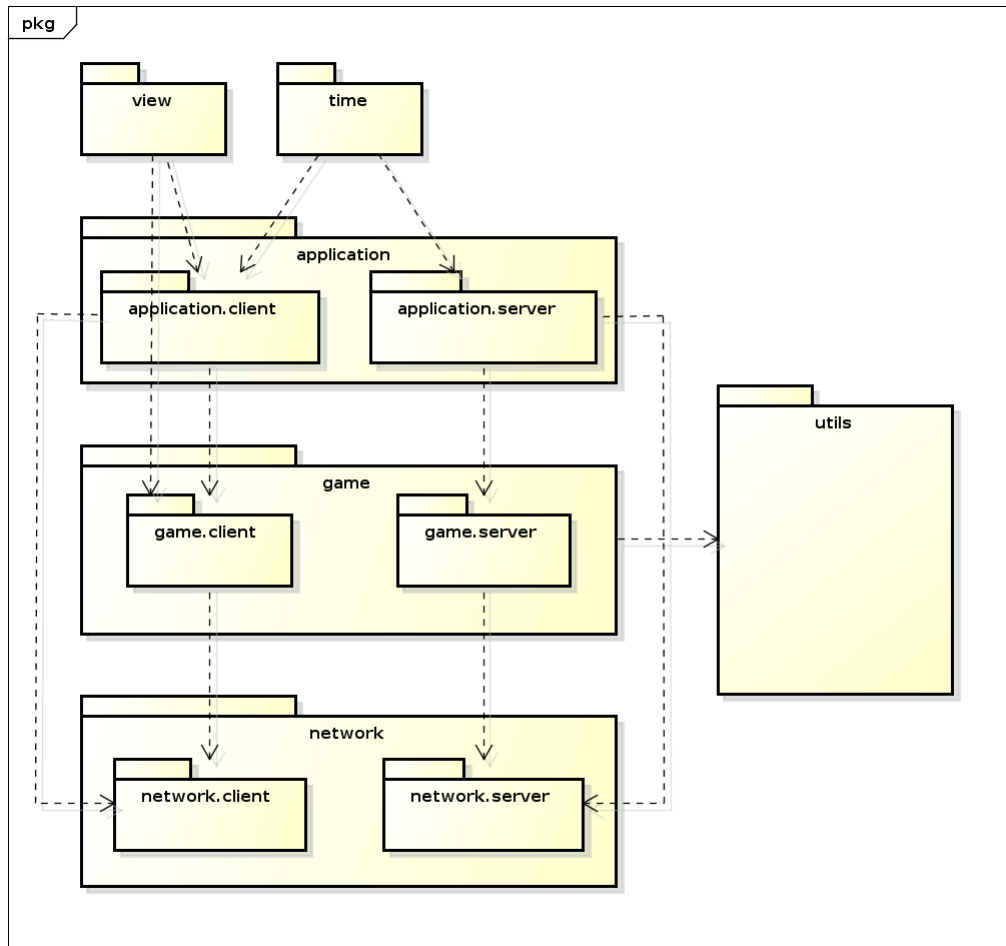
<Beschreibt die Softwarearchitektur eines Systems und wie sie sich präsentiert (am besten mit einem Bild um eine Übersicht zu ermöglichen) und einzelne Beschreibungen zu den einzelnen Elementen des Systems>

## 4 Architektonische Ziele & Einschränkungen

<Beschreibt die Softwareanforderungen und Objekte, welche einen Einfluss auf die Architektur haben (z.B. Safety, Security, Privacy, Distribution, usw.); Beinhaltet auch eine Beschreibung von Design und Implementationsstrategie, Entwicklungstools, usw.>

## 5 Logische Architektur

Dieses Package Diagramm zeigt sowohl Client, als auch Server. Client und Server sind zwei eigenständige Applikationen, die getrennt ausgeführt werden. Sie verwenden jedoch teilweise die gleichen Packages.

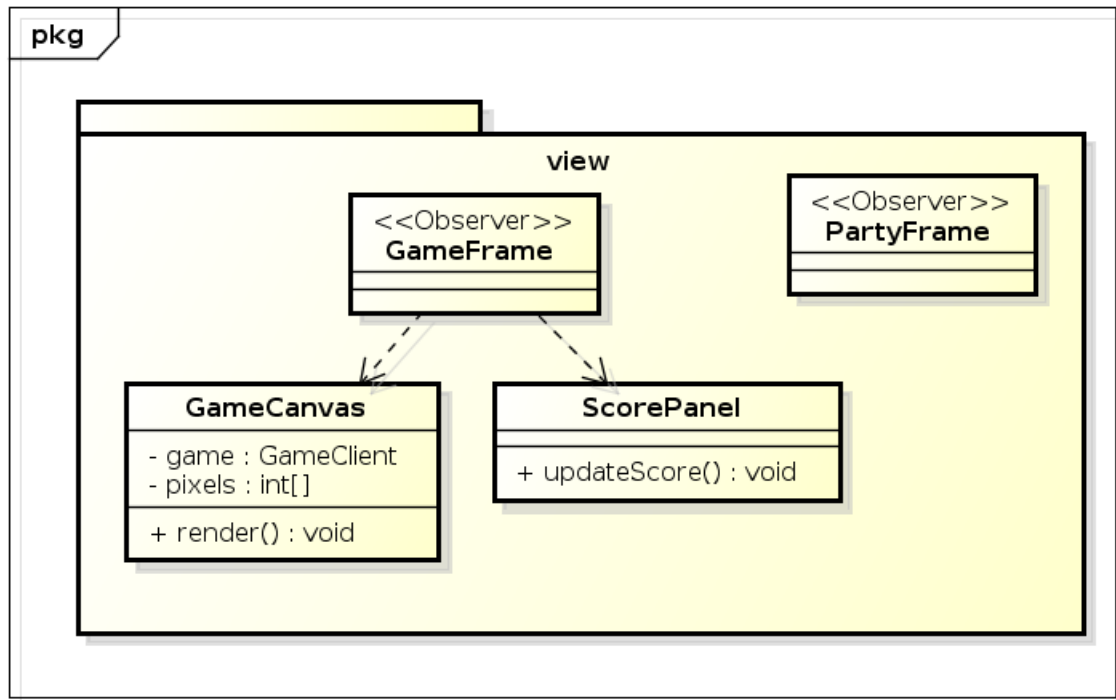


powered by Astah

## 5.1 <Presentation/view>

Im Package view befinden sich Frames und Canvas, die für die Presentation des Clients notwendig sind.

### 5.1.1 Klassenstruktur



powered by Astah

#### GameFrame

Der GameFrame ist ein Observer und delegiert die Notifies an das zugehörige Panel, die sich dann selber auf den neusten Stand bringen.

#### GameCanvas

Der GameCanvas kümmert sich nur um das Rendering, also das Zeichnen der Szene. Dabei delegiert er jedoch nur die pixels[] an alle Sprites, welche sich dann eigenständig zeichnen. Der GameCanvas kümmert sich dann um das performante Buffering.

Methode	Beschreibung
render():void	Kümmert sich um die BufferStrategy und delegiert das zeichnen der Sprites direkt an die Sprites selbst.

### **5.1.2 Wichtige interne Abläufe**

<Beschreibung von wichtigen internen Abläufen>

## **5.2 Wichtige Abläufe**

<Beschreibung von wichtigen Abläufen (packageübergreifend)>

## **6 Prozesse und Threads**

<Wenn mehrere Prozesse oder Threads eingesetzt werden wird hier beschrieben, wie diese ablaufen, miteinander funktionieren, Daten austauschen, sich synchronisieren, usw.>

## **7 Deployment**

<Beschreibung der einzelnen Komponenten und deren Aufteilung (auf welchen Umgebungen, Servern, usw. laufen die Komponenten)>

## **8 Datenspeicherung**

<Beschreibung mit Diagramm der Datenspeicherung (Datenmodell, z.B. Datenbank)>

## **9 Grössen und Leistung**

<Einschränkungen der Applikation bezüglich Speicher, Leistung, etc. . . . (zum Beispiel: Verwaltung unterstützt maximal 20'000 Einträge)>